AD-A040 772 HARRIS CORP MELBOURNE FLA ELECTRONIC SYSTEMS DIV F/G 20/6
FIBER OPTICS DESIGN AID PACKAGE.(U)
MAY 77 I B SLAYTON, D J ESTAPA, J R JONES F30602-76-C-0246

UNCLASSIFIED RADC-TR-77-163 NL

1 OF 3

AD A040772

ADA040772

(12)

RADC-TR-77-163
Final Technical Report
May 1977

FIBER OPTICS DESIGN AID PACKAGE

Harris Corporation

Approved for public release; distribution unlimited.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report contains a large percentage of machine-produced copy which is not of the highest printing quality but because of economical consideration, it was determined in the best interest of the government that they be used in this publication.

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

This report has been reviewed and is approved for publication.

APPROVED:

PAUL F. SIERAK
Project Engineer

APPROVED:

FRED I. DIAMOND
Technical Director
Communications and Control Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-77-163	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FIBER OPTICS DESIGN AID PACKAGE		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) I. B. Slayton C. R. Patisaul D. J. Estapa J. R. Jones		8. CONTRACT OR GRANT NUMBER(s) F30602-76-C-0246
9. PERFORMING ORGANIZATION NAME AND ADDRESS Harris Corp/Electronic Systems Division Melbourne FL 32901		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 65231104
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (DCCW) Griffiss AFB NY 13441		12. REPORT DATE May 1977
		13. NUMBER OF PAGES 196
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer:
Paul F. Sierak (DCCW)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Fiber Optics
Communications
Computer Modeling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report contains a computer based program which will provide for system level and component level performance analysis for point to point fiber optic communication links. It enables the user to optimize various aspects of a fiber optic communication link for either analog or digital transmission.

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408 972

LB

TECHNICAL REPORT SUMMARY

FIBER OPTICS DESIGN AID PACKAGE

This study Final Report provides documentation for a Fiber Optics Design Aid Package (FODAP). FODAP is intended to be a supplement to the RADC-Sponsored Optical Cable Communications Study (Contract Number F30602-74-C-0193). It provides the designer of optical fiber communications systems a method to facilitate his design procedures. This report consists mainly of a User's Guide to the FODAP computer software. This guide contains sections which discuss data definition and entry, FODAP module descriptions, a description of FODAP module interactions, a description of miscellaneous command statements, execution instructions. Also included is an input/output variable dictionary and a run listing. The User's Guide is supplemented by a Design Handbook which summarizes the design conditions and equations used in FODAP. Also included in the report is a section containing a series of design curves generated using the FODAP software. These curves can be used to examine trade-offs at the component or systems level.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

EVALUATION

This effort has provided a computer based program that will allow a communication system designer to analyze fiber optic communication links at either a system level or component level. This effort is based on a previous RADC effort titled "Optical Cable Communications Study" under which a graphical design package was developed for fiber optic analysis. The present program computerized and updated those results so that the fiber optic link designer will be able to perform tradeoffs and parameter optimization in an efficient manner.

The design tool developed under this program will aid in providing the fiber optic technology update for the services.

Paul Sierak

PAUL F. SIERAK
Project Engineer

SECTION 1.0

INTRODUCTION

1.0 INTRODUCTION

This document is a result of a nine-month program of study (Contract Number F30602-76-C-0246) conducted by Harris Corporation's Electronic Systems Division for the Rome Air Development Center. The objective of this study was to supplement the RADC sponsored Optical Cable Communications Study (F30602-74-C-0193) with a Fiber Optics Design Aid Package (FODAP) to facilitate the design of optical fiber communications systems. This package consists of a design handbook which outlines the procedures used in optical fiber systems design, a user's guide to the computer software used in the study and a series of design curves generated using the FODAP software.

In order to make FODAP most useful to the systems designer, this report has been partitioned into three sections. Section 2 consists of a design handbook. In this section, the design philolosphy and design procedures used in FODAP are outlined. Terminology is also presented and the design equations are summarized. Section 3 contains a user's guide to the computer software. It explains the procedures for operating FODAP including file maintenance, execution of the modules and subroutines which FODAP contains, input data requirements, data definition libraries and examples. Section 4 contains a compilation of design curves which are the useful final output of FODAP. These curves can be used by the systems engineer to design the individual system components such as the transmitter, receiver and fiber cable; or they can be used to design total systems using general classes of component types.

SECTION 2.0

DESIGN HANDBOOK FOR FODAP

2.0 DESIGN HANDBOOK FOR FODAP

2.1 Introduction

To make the best use of FODAP, the designer should have a working knowledge of the design philosophy and terminology used in constructing the computer software associated with FODAP. Such a knowledge will allow the designer to interpret the results and better understand the component-level and systems-level design tradeoffs. This section is intended to serve as such an introduction.

In FODAP, the design of an optical fiber communications system is done at two levels. One may design at the individual component level, i.e., one may design transmitters, receivers or select fiber type. Or, because of the interaction of components in an optical fiber link, one may design at the system level. In the sections that follow, the design approach, including equations and terminology is explained for each level of design. In particular, the information pertaining to receiver design is discussed in Section 2.2, fiber design relations are discussed in Section 2.3, Section 2.4 contains the transmitter design information and Section 2.5 describes the system-level design information. Each section is discussed from the point of view of the designer who is using FODAP.

It should be pointed out that the information contained in this section is intended to be a summary of design information. Details such as the derivation of design equations are beyond the scope of this work. For such information, the reader is referred to the Optical Cable Communications Study[1] which was mentioned earlier.

2.2 Receiver Analysis

It is possible to configure a number of different fiber optic communication systems using a variety of optical and electrical modulation techniques. In virtually all practical optical cable systems, intensity modulation (IM) of the optical source is employed together with direct detection at the receiver. Intensity modulation refers to varying the average (relative to an optical carrier cycle) output power of the optical source in proportion to an electrical stimulus, e.g., the drive current of an LED or ILD may be varied to cause changes in the optical power output. Direct detection at the optical receiver is accomplished through the use of a device such as a photodiode which converts average optical power (again relative to an optical carrier cycle) to an electrical signal.

An optical source may be intensity modulated in either continuous (analog) or discrete (digital) fashion, depending on system requirements. The electrical signal which modulates the optical source may, in term, be modulated in any one of a number of ways. System performance (signal reproduction fidelity) is largely determined by the noise characteristics of the optical receivers and electrical demodulators used, and by the response of the optical detector to the incident optical signal. FODAP is capable of theoretical predictions of optical receiver performance for a number of modulation types. This capability resides in the Receiver Analysis Module (RAM).

FODAP characterizes modulation types according to whether the input information signal is analog or digital and whether the input signal directly intensity modulates the optical source or first undergoes an electrical modulation. Systems in which the input electrical signal directly modulates the optical source are called baseband systems. If an intermediate electrical modulation is used, the system is referred to as a subcarrier system.

The RAM provides FODAP with the ability to predict system performance in terms of output signal-to-noise ratio (SNR) for analog input signals and output bit error rate (BER) for digital input signals. Then performance predictions are based on analytical expressions which were derived in Reference 1 and which are summarized in the subsequent

paragraphs. The analog transmission formats considered are baseband analog IM, and IM using AM, SSB-SC, DSB-SC or FM electrical subcarriers. The digital techniques included in FODAP are baseband on-off keying (OOK), baseband binary pulse-position modulation (BPPM) and IM using FSK or PSK electrical subcarriers. Other modulation schemes which may be regarded as analog-digital hybrids have been studied but not included in FODAP, e.g., analog pulse-position modulation[2,3] (PPM) and pulse frequency modulation[4] (PFM). While analog PPM may find application in certain specific systems, the associated implementation complexity and system criticality[5] precludes, for the present time at least, its wide spread use. Optical PFM appears more promising and doubtless will receive more attention in the future.

In the paragraphs which follow basic receiver parameters are defined and the mathematical expressions used by the RAM to predict receiver performance are summarized. It is assumed throughout that the reader has an understanding of standard electrical modulation and demodulation techniques and is acquainted with the basics of optical cable communication systems using direct detection optical receivers.

2.2.1 Assumed Receiver Structure and Parameter Definitions

The receiver structure assumed by the RAM is shown in Figure 2.2.1. Optical-to-electrical conversion is provided by a photodiode which produces a photocurrent, i_p, proportional to the intensity of the impinging optical signal. The photocurrent is corrupted by additive noise, i_n, which is conveniently represented as being referred to the amplifier input. The amplifier drives an electrical demodulator or detector appropriate to the modulation format chosen. This demodulator may be as simple as a noise rejection filter in the case of baseband analog IM or as elaborate as an FDM demultiplexer in the case of a system using multiple electrical subcarriers to intensity modulate the optical carrier.

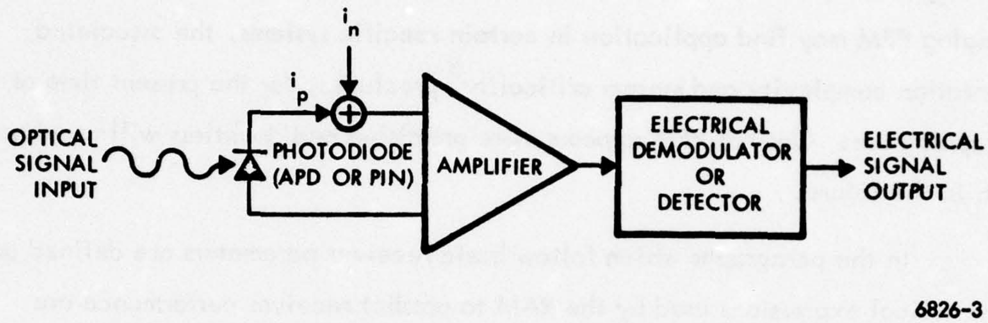


Figure 2.2.1. Receiver Structure

Photodiode parameters required by the RAM are:

Unity gain responsivity - r

Avalanche gain - G

Avalanche gain excess noise exponent - α_D

Surface (nonmultiplied) leakage current - I_L

Bulk (multiplied) leakage current - I_b

This photodiode description may be specialized to the case of a PIN photodiode (no avalanche gain) by setting G = 1 and $I_b = 0$. The value used for α_D in such a situation is arbitrary.

The noise current, i_n, is made up of several components. These components are contributed by the photodiode, the amplifier and by the quantum nature of the received optical signal itself. For the purpose of predicting optical receiver performance, the noise contributions are generally described by their second order statistics, namely the input-referred mean-square noise currents. The major noise sources considered by the RAM are:

Quantum noise: $\langle i_Q^2 \rangle = 2qrG^{2+\alpha_D} P_R b$

Dark current shot noise: $\langle i_D^2 \rangle = 2qb\,(I_L + G^{2+\alpha_D} I_b)$

Input referred amplifier noise: $\langle i_A^2 \rangle$

In the expressions above

q = charge on an electron (1.6×10^{-19} C)

P_R = average (long-term) optical carrier power incident on the photodiode

b = noise-equivalent bandwidth of the optical receiver (or receiver channel in the case of an FDM system using multiple electrical subcarriers)

There are as many expressions for the amplifier noise current as there are amplifier designs. For this reason it was described that the calculation of the amplifier noise performance should be an "outboard" computation to FODAP. As a precautionary note, the reader is reminded that $< i_A^2 >$ must be calculated for the particular bandwidth of interest. Some amplifiers for fiber optic receivers have significant non-white noise components. In the interest of computational simplicity, however, some of the RAM's analytical expressions are based on the assumption that the noise is spectually flat over the desired bandwidth. Therefore, interpretation of RAM results should be tempered with appreciation of these simplifying assumptions.

Other general parameters required by the RAM are:

Intensity modulation index - M

Digital data rate - R

RMS value of the analog signal - $\sqrt{< s^2 >}$

It is assumed that the analog signal is normalized such that

$$\left| s(t) \right| \leq 1$$

The performance measures used by the RAM are signal-to-noise ratio in the case of analog inputs and bit error rate for digital inputs. Signal-to-noise ratio is defined as the ratio of mean-square signal-to-mean-square noise as measured in the message bandwidth at the receiver output. Bit error rate is just the probability of a bit error at the receiver output assuming equiprobable occurrences of ONE's and ZERO's at the input.

2.2.2 Performance Equations

The paragraphs which follow summarize the equations used by the RAM to compute receiver performance.

2.2.2.1 Baseband Analog IM

The output SNR for a direct detection fiber optic receiver with baseband analog IM input is

$$\mathrm{SNR} = \frac{(MrGP_R)^2 < s^2 >}{< i_Q^2 > + < i_D^2 > + < i_A^2 >} \quad (2.2.2.1\text{-}1)$$

The noise terms are defined in Section 2.2.1.

2.2.2.2 Subcarrier Analog IM

The SNR expressions for the subcarrier analog IM receivers are:

$$\mathrm{SNR}_{AM} = \frac{(M_{SC} M_a rGP_R)^2 < s^2 >}{2 \left[< i_Q^2 > + < i_D^2 > + < i_A^2 > \right]} \quad (2.2.2.1\text{-}2)$$

$$\mathrm{SNR}_{DSB\text{-}SC} = \frac{(M_{SC} rGP_R)^2 < s^2 >}{2 \left[< i_Q^2 > + < i_D^2 > + < i_A^2 > \right]} \quad (2.2.2.1\text{-}3)$$

$$\mathrm{SNR}_{SSB\text{-}SC} = \frac{(M_{SC} rGP_R)^2 < s^2 >}{< i_Q^2 > + < i_D^2 > + < i_A^2 >} \quad (2.2.2.1\text{-}4)$$

$$\mathrm{SNR}_{FM} = \frac{3 \beta^2 (M_{SC} rGP_R)^2 < s^2 >}{2 \left[< i_Q^2 > + < i_D^2 > + < i_A^2 > \right]} \quad (2.2.2.1\text{-}5)$$

In the above expressions, the notation M_{SC} is used to indicate the intensity modulation index of the individual subcarrier signal, i.e., it is not the overall, composite intensity modulation index of the optical source in the case of multiple subcarriers. M_a is the amplitude modulation index of the electrical AM subcarrier. The reader is reminded that care must be exercised in selecting M_{SC} and M_a for an AM subcarrier system in order to avoid overmodulating the optical source on signal peaks.[1] β is the frequency modulation index of the FM subcarrier.

All of the subcarrier IM SNR expressions presented above assume that the amplifier noise has a flat spectrum and that all of the noise component are measured in the baseband message bandwidth.

2.2.2.3 Baseband Digital OOK and BPPM

The RAM computes BER for baseband digital OOK and BPPM according to

$$\mathrm{BER} = \mathrm{erfc}\left\{\frac{\sqrt{2}\,\frac{4}{\pi}\,rGP_R}{\sqrt{2<i_Q^2> + <i_D^2> + <i_A^2>} + \sqrt{<i_D^2> + <i_A^2>}}\right\} \qquad (2.2.2.3\text{-}1)$$

where

$$\mathrm{erfc}\left\{x\right\} = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-y^2/2}\, dy$$

This result is based on the use of the optimum bit decision threshold.[1] Noise components are computed as given in Section 2.2.1 with

$$b = \frac{\pi}{3} \cdot \frac{R}{2} \qquad \text{(OOK)} \tag{2.2.2.3-2}$$

$$b = \frac{\pi}{3} \cdot R \qquad \text{(BPPM)} \tag{2.2.2.3-3}$$

Required optical power for a given BER is obtained by solving for P_R in the BER expression, yielding

$$P_R = Q^2 \, \frac{\pi^3}{24} \, \frac{q}{r} \, G^{\alpha_D} b + \frac{Q}{\frac{4}{\pi} \, \frac{1}{\sqrt{2}} \, rG} \sqrt{2qb\,(I_L + I_b G^{2+\alpha_D}) + \langle i_A^2 \rangle}$$

(2.2.2.3-4)

where

$$\text{BER} = \text{erfc} \left\{ Q \right\} \tag{2.2.2.3-5}$$

2.2.2.4 Subcarrier Digital IM

Subcarrier digital IM performance is given by

$$\text{BER}_{PSK} = \text{erfc} \left\{ \sqrt{\frac{2E_b}{N_o}} \right\} \tag{2.2.2.4-1}$$

$$\text{BER}_{FSK} = \text{erfc} \left\{ 1.1 \sqrt{\frac{E_b}{N_o}} \right\} \tag{2.2.2.4-2}$$

with

$$\frac{E_b}{N_o} = \frac{\frac{1}{2}\,(M_{SC}\,rGP_R)^2}{\langle i_Q^2 \rangle + \langle i_D^2 \rangle + \langle i_A^2 \rangle} \qquad (2.2.2.4\text{-}3)$$

The noise terms are computed using

$$b = R \qquad (2.2.2.4\text{-}4)$$

The amplifier noise contribution is assumed to be spectually flat and matched filter detection is assumed in the electrical demodulators.

2.2.2.5 Avalanche Gain Optimization

Because of the dependence of both signal power and diode noise power on avalanche gain, there exists an optimum gain value which maximizes SNR for a given received optical power for analog channels or minimizes BER for a given received power for digital channels. In the case of baseband analog IM and both digital and analog subcarrier IM, this optimum gain can be obtained analytically and is

$$G_{opt} = \left[\frac{2\left\{2qI_L b + \langle i_A^2 \rangle\right\}}{\alpha_o\left\{2qb\,(rP_R + I_b)\right\}}\right]^{1/(2+\alpha_D)} \qquad (2.2.2.5\text{-}1)$$

The concept of optimum gain for multiple subcarriers is meaningful only if the amplifier noise spectrum is flat and, hence, G_{opt} has no bandwidth dependence.

Unfortunately, in the case of baseband digital systems, analytical expressions for optimum gain do not, in general, exist and the gain must be optimized numerically.[6] FODAP has the capability to perform this numerical optimization procedure.

2.3 Fiber Analysis

As with conventional cable systems, two of the main performance parameters of the optical fiber cable are attenuation or loss and bandwidth. A third important performance measure is the amount of optical power that can be launched into the fiber cable by the optical transmitter.

The computation of loss for a length of optical cable is just a straight forward multiplication of the cable length by the optical attenuation of the cable in dB per unit length.

The amount of power coupled into a fiber or fiber bundle depends strongly on both the characteristics of the fiber and the optical source. For this reason the assignment of coupled power calculation to either the Fiber Analysis Module (FAM) or to the Transmitter Analysis Module (TAM) of FODAP is somewhat an arbitrary choice. As it turns out, both the FAM and the TAM have the capability to compute coupled power. However, since it appears that in the future most optical sources for fiber communication will have a fiber stub or pigtail permanently attached, it seems most appropriate to compute coupled power in the TAM. The FAM determines coupled power only when no fiber pigtail is used on the transmitter. The reader is referred to Section 2.4, which deals with the TAM, for details relating to coupled power computations.

By far the bulk of the computation done in the FAM is related to fiber cable bandwidth. Fiber bandwidth is limited by the temporal spreading or dispersion of optical energy as it propagates along a length of fiber waveguide. In multimode fibers, the only ones currently of practical significance, dispersion may be separated into two contributions. The first of these is intermodal dispersion which results from differing transmission delays for the various waveguide modes. Intermodal dispersion is often called simply modal dispersion. The second contribution is intramodal dispersion or the variation of propagation delay of optical energy at different wavelengths within a given waveguide mode. The dominant source of intramodal dispersion is the dependence of the index of refraction of the waveguide material on wavelength, or the material

effect. Thus intramodal dispersion is dependent on the optical source being used, in particular, on the spectral width of the source. Intramodal dispersion is sometimes called material dispersion. We prefer to call it spectral dispersion.

Multimode optical waveguides generally available today can be devided into two classes - step index and graded index. In a step index fiber the core material has a radially uniform index of refraction with an abrupt reduction or step change in refractive index at the core-cladding interface. A graded index fiber, on the otherhand, has a core index profile which decreases smoothly with radial distance from the fiber's longitudinal axis. There is no sharply defined core-cladding interface in a graded index fiber. Optical energy propagation along a fiber waveguide is supported by total internal reflection at the core-cladding interface in the case of a step index while continuous refraction or bending of rays provides "guidance" in a graded index fiber. As will be shown in example FAM computations which appear later in this report, the effect of index grading is a substantial reduction in intermodal dispersion.

A functional description of the core index profile which can be used to describe both step and graded index fibers is

$$n(r) = n_1 \left[1 - 2\Delta \, (r/2) \, \alpha \right]^{1/2}, \quad r < 2 \qquad (2.3\text{-}1)$$

$$n(r) = n_2, \quad b \geq r \geq a$$

where

n_1 = core index of refraction on the longitudinal axis of the fiber

n_2 = cladding index of refraction

r = *radial distance from the fiber axis*

a = core radius

b = fiber radius (core + cladding)

$\Delta = (n_1^2 - n_2^2)/2n_1^2 \approx (n_1 - n_2)/n_1$ for small Δ

α = fiber index profile shape parameter

In the limit of $\alpha \rightarrow \infty$, equation (2.3-1) describes an ideal step index fiber. Figure 2.3 shows plots of the index profiles of an ideal step index fiber ($\alpha = \infty$), a practical step index fiber ($\alpha = 25$) and a "near parabolic" graded index fiber ($\alpha = 2$).

As will be shown subsequently, both intermodal and intramodal dispersion depend directly on the index difference parameter Δ. This parameter is usually specified in terms of the numerical aperture (NA) of the fiber which is defined as

$$NA = \sin\theta_A \approx n_1\sqrt{2\Delta} \qquad (2.3\text{-}2)$$

Here θ_A is the acceptance half-angle of the fiber, i.e., the maximum angle relative to the fiber axis at which plane wave optical energy impinging on the end face of the fiber will be accepted and propagated. Plane wave excitation striking the fiber end at angles greater than θ_A passes through the core into the cladding and is lost. Thus the NA is an indicator of the amount of optical power that can be launched into the fiber, and, through its direct relation to Δ, is a measure of the dispersiveness of the waveguide.

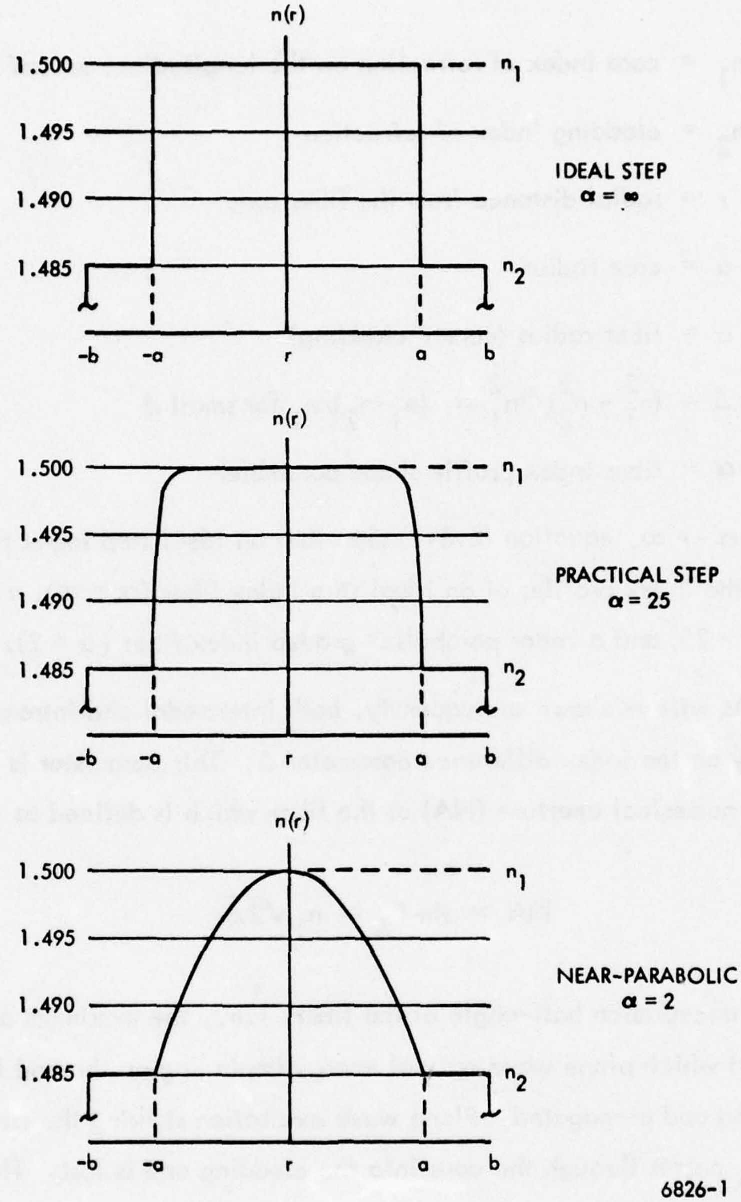


Figure 2.3. Refractive Index Profiles for Multimode Fibers ($n_1 = 1.5$, $\Delta = 0.01$)

The next five subsections present the equations used by the FAM to predict dispersion and bandwidth for general multimode optical fibers described by equation (2.3-1). Dispersion predictions are taken from recent work by Olshansky and Keck.[7,8]

2.3.1 Intermodal Dispersion

It has been found that a convenient measure of fiber dispersion for baseband digital links is the rms duration of the fiber's optical intensity impulse response given by

$$\sigma = \left[\frac{1}{A} \left\{ \int_{-\infty}^{\infty} t^2 h(t)dt - \left(\int_{-\infty}^{\infty} th(t)dt \right)^2 \right\} \right]^{1/2} \tag{2.3.1-1}$$

$$A = \int_{-\infty}^{\infty} h(t)dt$$

where h(t) is the optical intensity impulse response of the fiber.

The intermodal dispersion of a fiber, ignoring mode coupling effects, is approximately

$$\sigma_{inter}(L) = \frac{LN_1\Delta}{2c} \frac{\alpha}{\alpha+1} \left(\frac{\alpha+2}{3\alpha+2}\right)^{1/2} \left[C_1^2 + \frac{4C_1C_2\Delta(\alpha+1)}{2\alpha+1} + \frac{4\Delta^2 C_2^2 (2\alpha+2)^2}{(5\alpha+2)(3\alpha+2)} \right]^{1/2} \tag{2.3.1-2}$$

c = free space speed of light

$$C_1 = \frac{\alpha - 2 - \epsilon}{\alpha + 2} \tag{2.3.1-3}$$

$$C_2 = \frac{3\alpha - 2 - 2\epsilon}{2(\alpha + 2)} \tag{2.3.1-4}$$

$$N_1 = n_1 - \lambda \frac{dn_1}{d\lambda} \tag{2.3.1-5}$$

$$\epsilon = \frac{-2n_1}{N_1} \frac{\lambda}{\Delta} \frac{d\Delta}{d\lambda} \tag{2.3.1-6}$$

λ = source wavelength

L = waveguide length

There is an optimum value of the index profile, α_{opt}, which minimizes the intermodal disperson. This value is

$$\alpha_{opt} = 2 + \epsilon - \Delta \frac{(4+\epsilon)(3+\epsilon)}{(5+2\epsilon)} \tag{2.3.1-7}$$

Note the linear dependence of intermodal dispersion on fiber length L, in (2.3.1-1). It is known, however, that for sufficiently long optical cables the intermodal dispersion has a $\sqrt{L}$ - dependence rather than linear dependence. This behavior is attributed to mode coupling or mixing within the waveguide caused by microstresses imposed on the fiber by the cabling material. Intermodal dispersion is reduced by mode coupling because optical energy propagates first in one mode and then in another, etc. The propagation delay experienced by the optical energy is then a weighted average of the delays of the various modes. This averaging of propagation time tends to reduce differential delay and, therefore, intermodal dispersion.

The asymptotic behavior of intermodal dispersion may be expressed by $\sigma_{inter}(L)$ for $L \ll L_c$ and $\sigma_{inter}(\sqrt{LL_c})$ for $L \gg L_c$. L_c is the mode coupling length of the fiber and defines the point of intersection of the linear and $\sqrt{L}$ asymptotes. Mode coupling within the fiber causes some propagating optical energy to be coupled into radiating modes or lossy cladding modes. Consequently, mode coupling introduces excess loss above the uncabled fiber loss. The mode coupling length of a cabled fiber may be inferred from a knowledge of this excess loss according to

$$L_c \approx \frac{k(\alpha)}{\ell_e} \qquad (2.3.1\text{-}8)$$

where $k(\alpha)$ is a cable parameter depending on the nature of the microstresses on the fiber and ℓ_{mc} is the excess loss per unit length due to male coupling. Figure 2.3.1 shows a typical plot of $k(\alpha)$ versus α. The FAM uses the information contained in this plot to predict L_c.

2.3.2 Intramodal Dispersion

The intramodal dispersion of a fiber is given approximately by

$$\sigma_{intra}(L) = \frac{L\sigma_\lambda}{c\lambda}\left[\left(-\lambda^2 \frac{d^2 n_1}{d\lambda^2}\right)^2 - 2\lambda^2 \frac{d^2 n_1}{d\lambda^2} N_1 \Delta \left(\frac{\alpha - 2 - \epsilon}{\alpha + 2}\right)\left(\frac{2\alpha}{2\alpha+2}\right)\right.$$

$$\left. + (N_1 \Delta)^2 \left(\frac{\alpha - 2 - \epsilon}{\alpha + 2}\right)^2 \frac{2\alpha}{3\alpha+2}\right]^{1/2} \qquad (2.3.2\text{-}1)$$

where

σ_λ = rms spectral width of the optical source

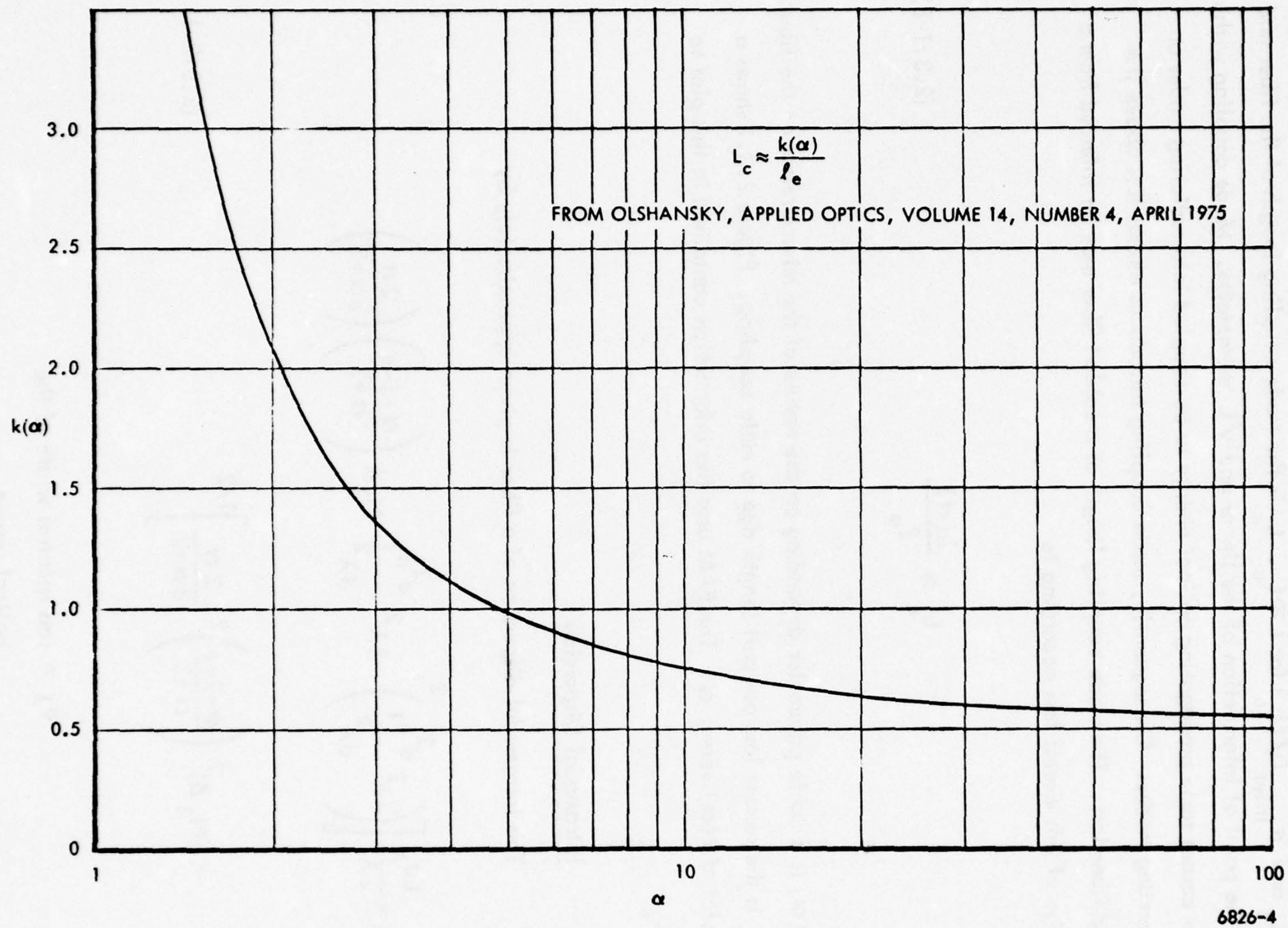


Figure 2.3.1. $k(\alpha)$ Versus α

2.3.3 Total Dispersion

Intermodal and intramodal dispersion combine in a "square-root of the sum of the squares" fashion so that the asymptotic behavior of the total fiber dispersion is

$$\sigma_{total}(L) = \left\{ \sigma^2_{inter}(L) + \sigma^2_{intra}(L) \right\}^{1/2} \quad , \; L \ll L_c$$

$$\sigma_{total}(L) = \left\{ \sigma^2_{inter}(\sqrt{LL_c}) + \sigma^2_{intra}(L) \right\}^{1/2} \quad , \; L >> L_c \tag{2.3.3-1}$$

2.3.4 Fiber Cable Bandwidth

The modulation frequency response of the fiber cable is obtained by Fourier transformation of the fiber's optical intensity impulse response. A good approximation for this impulse response for long cables is a Gaussian shape

$$h(t) = \exp \left\{ \frac{-t^2}{2\,\sigma^2_{total}(L)} \right\} \tag{2.3.4-1}$$

Taking the Fourier transform yields

$$\left| H(f) \right| = \exp \left\{ -2\pi^2 \, \sigma_{total}(L) \, f^2 \right\} \tag{2.3.4-2}$$

The frequency response has been normalized to unity at $f = 0$ and represents the equivalent frequency response of the fiber cable to the modulating signal, i.e., the equivalent electrical amplitude response due to the cable of the modulating signal after direct detection by the photodetector is given by equation (2.3.4-2). Hence the 3 dB modulation bandwidth of the optical cable is obtained from equation (2.3.4-2) and is

$$f_{3\ dB}(L) = \frac{0.132}{\sigma_{total}(L)} \tag{2.3.4-3}$$

The use of the Gaussian impulse response assumption to arrive at $f_{3\ dB}(L)$ given in equation (2.3.4-4) leads to conservative results, at least in the rms bandwidth sense. The validity of this assertion is derived from the uncertainty principle of the Fourier transform[9] which states that of all time signals having rms duration σ, the one having the smallest rms bandwidth is *the Gaussian pulse*.

2.4 Transmitter Design

As far as FODAP is concerned, the design of a transmitter consists of selection of the desired optical source and then specifying the pertinent parameters of that source. Design of optical source modulation circuitry is beyond the scope of FODAP. Source selection is governed by performance and cost considerations. This process is performed using the Transmitter Analysis Module (TAM) of FODAP. There are several source types available and they fall into four basic categories:

1. edge-emitting light emitting diode (LED)
2. surface-emitting LED
3. injection laser diode (ILD)
4. other laser types including helium-neon (He-Ne) lasers and neodymium-dopped yttrium-aluminum-garnet (Nd:YAG) lasers

The most common types of sources used are the surface emitting LED and ILD. To fully specify the source used, one must specify whether a fiber pigtail is attached to the source since in FODAP, the pigtail is considered to be part of the transmitter. A pigtail is attached in the cases where the optical alignment between source and fiber is extremely critical. This is usually the case for sources with small emitting areas. Some manufacturers will supply their sources with pigtails attached (for instance, Bell Northern Research). In addition, a pigtail may be attached in cases where EMI shielding is important. With a pigtail attached, the source may be located inside a shielded box with only a small hole through which the pigtail passes.

The important source parameters as required by FODAP are:

1. The optical power available for coupling to the pigtail or fiber cable.
2. The spacial distribution of the available power including the size of the optical source.

3. The spectral properties of the source including the peak-emitting wavelength λ and the rms spectral bandwidth σ_λ.

4. The properties of the fiber pigtail (if used).

5. The modulation frequency response or light output risetime.

Some of these parameters such as available optical power and spectral properties are functions of temperature. If that is the case, then they are specified at the temperature of interest.

The details of how these parameters are specified and what calculations are performed in the transmitter section of FODAP using these parameters are discussed for the various source types in the sections that follow.

2.4.1 Surface Emitting LED

2.4.1.1 Available Optical Power

The optical power available from such a source for coupling to the fiber cable in a system is an important parameter. For the case where a fiber pigtail is used, the available power from the transmitter that is inputted to the other modules of FODAP is the power captured by the pigtail. In order to calculate this power, one must evaluate the coupling losses. The two main loss sources which are accounted for in TAM are the numerical aperature losses and the unintercepted illumination losses. The former arises from mismatches between the fiber pigtail acceptance cone and LED emission cone and the latter results from mismatches between fiber core area and LED emission area. In order to evaluate these losses, one must know the LED emission angle, i.e., the angle at which the intensity pattern mills; the fiber acceptance angle, i.e., the numerical aperature of the fiber; the source radiating area and the fiber core area. The numerical aperature loss is calculated using

$$L_{NA}\,(\mathrm{dB}) = 10 \log \left[\frac{\dfrac{\cos\left[(n-1)\,\theta_{NA}\right]}{n-1} - \dfrac{\cos\left[(n+1)\,\theta_{NA}\right]}{n+1} - \dfrac{2}{n^2-1}}{\dfrac{\cos\left[\dfrac{(n-1)}{n}\,\dfrac{\pi}{2}\right]}{n-1} - \dfrac{\cos\left[\dfrac{(n+1)}{n}\,\dfrac{\pi}{2}\right]}{n+1} - \dfrac{2}{n^2-1}} \right] \tag{2.4.1.1-1}$$

In this equation, $n = 90/\theta_n$ where θ_n is the null angle (Figure 2.4.1.1-1 illustrates this) and θ_{NA} is the fiber acceptance angle which is given in terms of fiber numerical operature, NA, by $\theta_{NA} = \sin^{-1}$ (NA). Note that the above result does not hold for n=1. That case is discussed below. Also, for $\theta_{NA} > \theta_n$, $L_{NA} = 0$ is used.

The unintercepted illumination loss is given by

$$L_{UI}\,(\mathrm{dB}) = 10 \log \frac{A_F}{A_s} \tag{2.4.1.1-2}$$

where A_F is the fiber core area and A_s is the LED emission area. Note that the above equation does not hold for $A_s < A_F$. There is no unintercepted illumination loss in that case.

To account for the case where a fiber bundle is attached as a pigtail, one must include the packing fraction loss. The packing fraction f_p is defined as the ratio of total core area to total area within fiber bundle termination. Figure 2.4.1.1-2 illustrates the packing fraction concepts. Hence, the packing fraction loss is

$$L_{PF}\,(\mathrm{dB}) = 10 \log (f_p) \tag{2.4.1.1-3}$$

When a single fiber is used as a pigtail, $f_p \equiv 1$ and $L_{PF} = 0$.

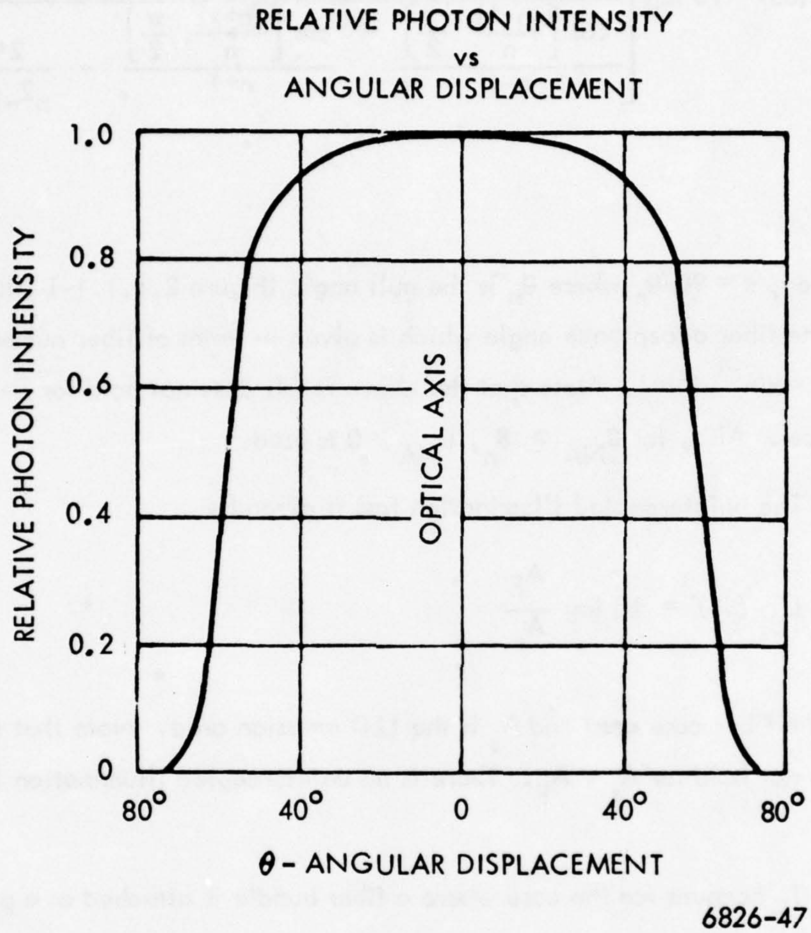


Figure 2.4.1.1-1. Plot of LED Output Intensity Versus Angle

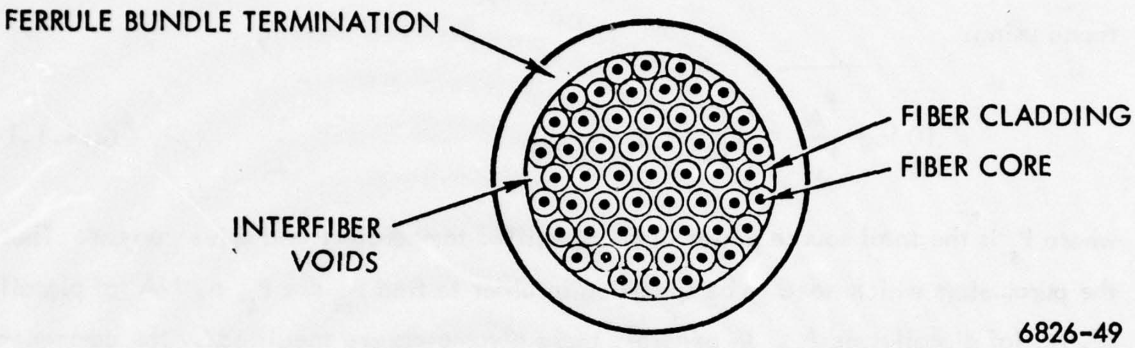


Figure 2.4.1.1-2. Packing Fraction Concepts

As to the special case noted above; when the null factor n=1, this describes a so-called Lambertian emitter. For this special case, the numerical aperature loss is given by

$$L_{NA}\ (\mathrm{dB}) = 10 \log (NA^2) \qquad (2.4.1.1\text{-}4)$$

In summary, the available power P_A for the case of an attached pigtail is found using:

$$10 \log \frac{P_A}{P_s} = L_{NA} + L_{UI} + L_{PF} \qquad (2.4.1.1\text{-}5)$$

where P_s is the total source power at the specified temperature and drive current. Thus, the parameters which need to be specified in order to find P_A are P_s, n, NA (of pigtail), f_p, A_F (of pigtail) and A_s. In general, these parameters are specified by the component manufacturer.

The above calculations are valid for the case where a step index fiber is used as the pigtail. If a graded index fiber is used that has the same NA and core area, an additional loss, L_{GI} = 3 dB must be added to equation (2.4.1.1-5) to account for the fact that fewer modes are captured by the graded index fiber[10]. However, at least one fiber manufacturer, Corning, accounts for this 3 dB loss by specifying an equivalent core area which is smaller than the actual area so as to increase the loss calculated by equation (2.4.1.1-2) (L_{UI}) by 3 dB. Thus, in the event of the use of a graded index fiber pigtail, the FODAP user should determine what method is to be used to account for L_{GI}.

For the case where a pigtail is not used and the LED is coupled directly to the fiber cable, P_A is simply P_s. In this case, the losses associated with coupling to the fiber cable are calculated in FAM of FODAP using the same equations as above.

In the event that the LED has an attached lens - whether it is an epoxy "dome" or glass lens integral with the LED package - the values used for the null factor n and area A_s are those at the lens output.

Most manufacturers specify the total source power, P_s. However, there are a few who specify either the source radiance, B (in W/sr-cm^2), or the radiant intensity, I_o (in mW/sr). For the case of the Lambertian emitter, the source power is derived from these parameters using the following relationships:

$$P_s = \pi I_o \qquad (2.4.1.1\text{-}6)$$

$$P_s = B \pi A_s \qquad (2.4.1.1\text{-}7)$$

2.4.1.2 Spectral Properties

As was pointed out in Section 2.3, the spectral properties of the source have a large impact on the overall system modulation capability. Since these are properties of the source, the spectral parameters λ, the peak emission wavelength and σ_λ, the rms spectral bandwidth are specified in TAM. However, no calculations are performed in TAM using these parameters, the information is passed onto FAM where calculations are made using λ and σ_λ. In general, these quantities are specified on manufacturers specification sheets. These parameters are functions of temperature and drive current level and so they are specified at the desired conditions. Figure 2.4.1.2-1 illustrates the spectral properties of typical LED and ILD sources.

2.4.1.3 Modulation Properties

The source modulation properties are important in determining the overall system modulation capability. Mose manufacturers specify the source light-output risetime, t_r. Others may specify the 3 dB modulation bandwidth, $f_{3\ dB}$. The two quantities are related using the following equation:

$$t_r = 0.35/f_{3\ dB} \qquad (2.4.1.3\text{-}1)$$

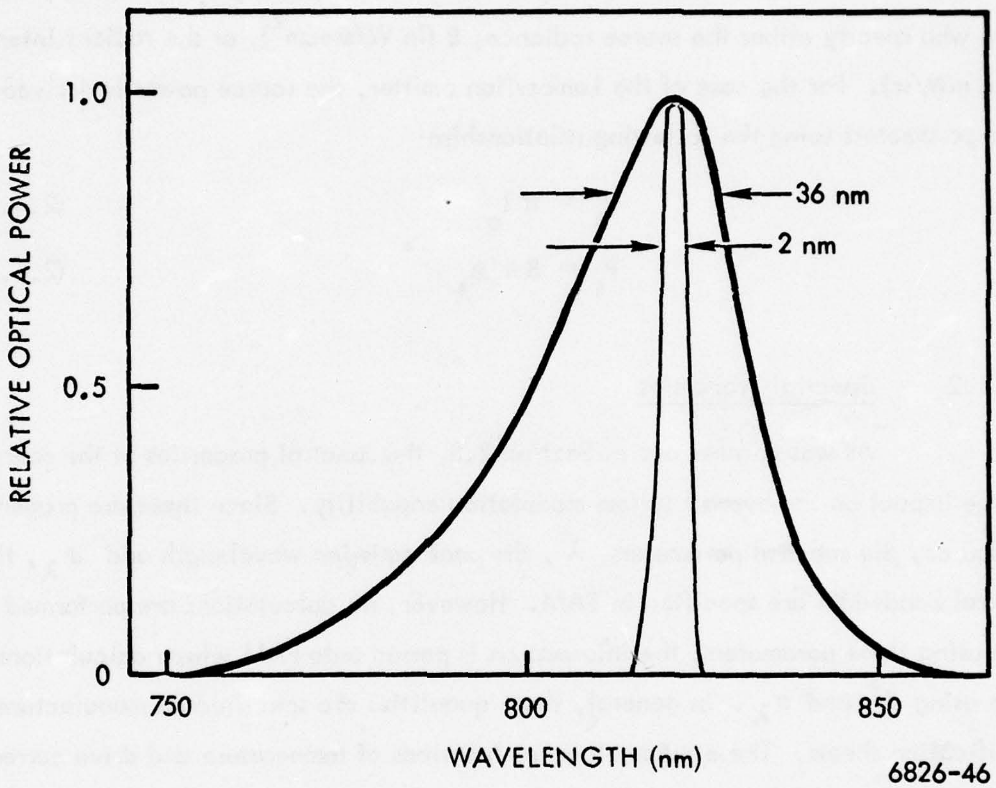


Figure 2.4.1.2-1. LED and ILD Spectral Properties

These parameters are functions of drive current level and drive circuit configuration and so they are specified for the appropriate conditions.

2.4.2 Edge Emitting LED

2.4.2.1 Available Optical Power

The relationships derived for the surface emitting LED apply for the case of cylindrical symmetry about the emitting axis. Edge emitting LED's have an eliptical emission pattern and this assymmetry makes them not ameanable to analysis. Hence, an empirical expression for P_A is used:[11]

$$P_A = 0.1\,P_s \left(\frac{NA}{0.14}\right)^2 \left(\frac{d_F}{90}\right)^2 f_p \qquad (2.4.2.1\text{-}1)$$

where NA is the pigtail or fiber numerical aperture (depending or whether a pigtail is used or not), d_F is the fiber or pigtail core diameter in microns and P_s is the average source power at the specified drive current and temperature.

In this case, if a graded index fiber is used as the pigtail, P_A as calculated above must be reduced by a factor of 2.

For the case where a pigtail is not used, P_A is simply P_s as before.

2.4.2.2 Spectral Properties

The FODAP user specified the spectral parameters λ and σ_λ as part of the TAM input. These data are commonly available from manufacturers specification sheets. They are specified by the FODAP user at the temperature and drive current levels of interest.

2.4.2.3 Modulation Properties

Either the light pulse risetime t_r or modulation bandwidth $f_{3\ dB}$ are specified using data supplied by manufacturers. They are user-specified taking into account the appropriate conditions.

2.4.3 ILD

2.4.3.1 Available Optical Power

The ILD is also an edge emitting device. Figure 2.4.3.1-1 illustrates the ILD emission pattern. The assymmetrical pattern makes exact calculation difficult - complicated ray tracing calculations are involved - and so empirical equations for P_A are used for the case where a pigtail is employed.[12] The coherence of the laser emission and the small radiating area allow the designer to couple considerably more power into a fiber than is possible with an LED. In addition, there are techniques for further enhancing the coupling. A common technique is the use of a microlens either attached to or melted on the fiber end.[13] In TAM for the case where a pigtail is used, P_A is calculated for the lensed or nonlensed situations using the equations below:

Nonlensed: $$P_A = 0.15\, f_p\, P_s \qquad (2.4.3.1\text{-}1)$$

Lensed: $$P_A = 0.5\, f_p\, P_s \qquad (2.4.3.1\text{-}2)$$

where P_s is the source power at the desired operating conditions. These equations are empirical but they are conservative so that their predictions are worst-case. In this particular case, the coupled power is the same whether the pigtail is a step or graded index fiber.

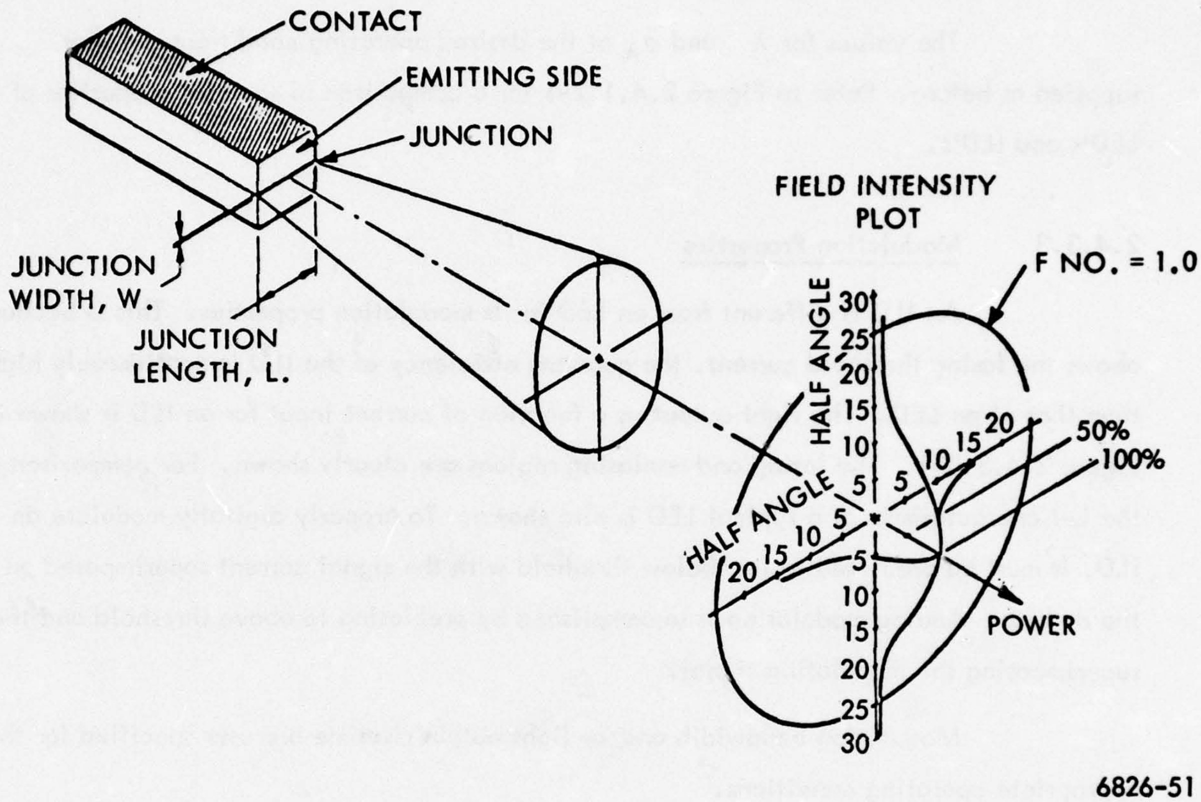


Figure 2.4.3.1-1. ILD Emission Properties

When no pigtail is used, P_A is simply P_s as before and coupling losses are calculated in FAM.

2.4.3.2 Spectral Properties

The values for λ and σ_λ at the desired operating conditions are user supplied as before. Refer to Figure 2.4.1.2-1 for a comparison of spectral properties of LED's and ILD's.

2.4.3.3 Modulation Properties

An ILD is different from an LED in its modulation properties. This is because above the lasing threshold current, the quantum efficiency of the ILD is considerably higher than that of an LED. The light output as a function of current input for an ILD is shown in Figure 2.4.3.3-1. The lasing and nonlasing regions are clearly shown. For comparison, the L-I characteristic of a typical LED is also shown. To properly digitally modulate an ILD, it must be prebiased to just below threshold with the signal current superimposed on the dc bias. Analog modulation is accomplished by prebiasing to above threshold and then superimposing the modulating signal.

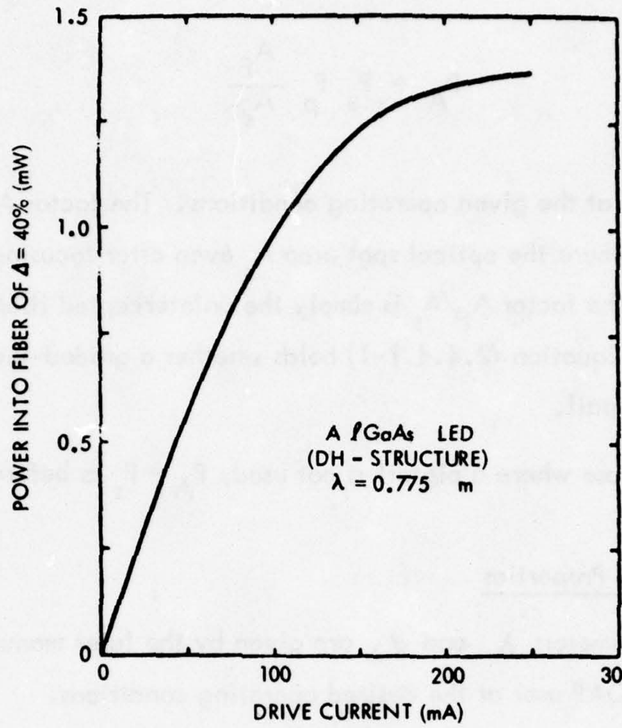
Modulation bandwidth and/or light output risetime are user specified for the appropriate operating conditions.

2.4.4 Other Laser Types

Other laser types may be used as optical sources. The two most common are the HeNe and Nd:YAG lasers.

2.4.4.1 Available Optical Power

Because of the coherence of the laser light, essentially all of the source power can be focused into an optical fiber. The only losses are those associated with reflections which are small enough to be ignored. Thus, for the case where a pigtail is used:



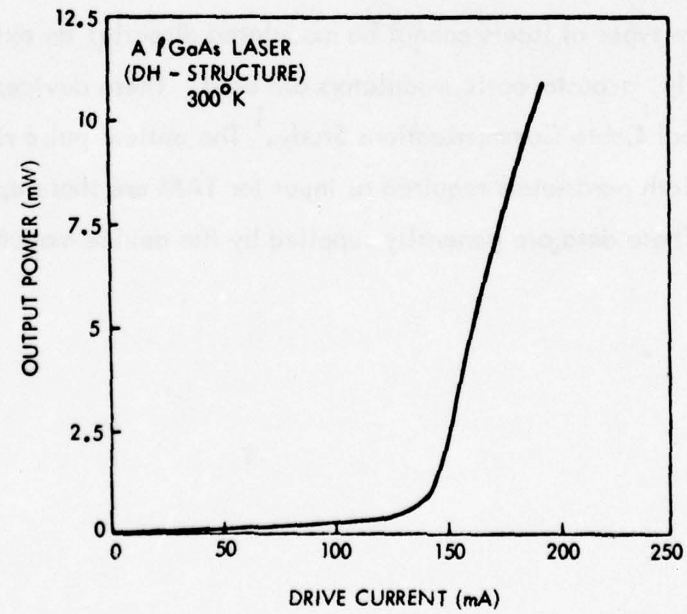


Figure 2.4.3.3-1. ILD and LED L-I Characteristics

$$P_A = P_s f_p \frac{A_F}{A_s} \qquad (2.4.4.1\text{-}1)$$

P_s is the source power at the given operating conditions. The factor A_F/A_s is included to account for the case where the optical spot area A_s even after focusing is larger than the fiber core area A_F. The factor A_F/A_s is simply the unintercepted illumination loss discussed in Section 2.4.1.1. Equation (2.4.4.1-1) holds whether a graded-index or step-index fiber is used as the pigtail.

In the case where a pigtail is not used, $P_A = P_s$ as before.

2.4.4.2 Spectral Properties

The parameters λ and σ_λ are given by the laser manufacturers. This data is inputted by the FODAP user at the desired operating conditions.

2.4.4.3 Modulation Properties

These types of lasers cannot be modulated directly; an external modulator is required. Typically, acousto-optic modulators are used. These devices are discussed in depth in the Optical Cable Communications Study.[1] The optical pulse risetime and/or modulation bandwidth parameters required as input for TAM are those appropriate to the modulator used. These data are generally supplied by the device manufacturers.

2.5 Systems Design

As was pointed out earlier, design of an optical fiber link occurs at two levels. First, the individual components must be designed or specified and then secondly, their interaction is considered in a system-level design. The overall design goals of a communication link involve the transfer of information from one point to another without degradation. For analog systems, this amounts to designing for a desired signal-to-noise ratio (SNR), bandwidth and distortion level. At the present state-of-the-art, analytical expressions for signal distortion in optical sources are not available. Thus, distortion effects are not considered in FODAP; they must be measured and specified independently. However, SNR and bandwidth analyses are performed by FODAP. For digital systems, the design goals are specified in terms of bit error rate (BER) and bandwidth. For both types of systems, the calculations that pertain to the overall system SNR or BER performance are performed in RAM. However, to summarize the interactions between modules, the Systems Analysis Module (SAM) gathers data from each module and displays it in terms of a link budget. The link budget concept is a very useful tool because it tells the designer whether or not he has "closed the link" in terms of having adequate optical power to meet the SNR or BER requirements. The link budget concept is discussed in more detail in Section 2.5.1 below. SAM also calculates an overall system bandwidth and displays it so that the designer will know if his link design has adequate signal bandwidth. Section 2.5.2 below describes this calculation.

2.5.1 Link Budget

To verify whether or not the design goals for SNR or BER have been met by the combination of the various component designs, the engineer can use a link budget. This link budget starts with the total source power. All of the system losses (the input coupling loss, the splice and connector losses, the fiber attenuation losses and the output coupling loss) are listed along with the received optical power. The received power required to meet the SNR or BER design goals is listed and the difference between this

number and the actual received power is noted. An example link budget is shown in Figure 2.5.1. This type of listing gives the designer a quick summary of whether or not his design goals have been met. It also helps him to see the weak points of his design. For example, because of source or fiber characteristics, a proposed design may suffer from overly large input and output coupling losses resulting in a power deficit. This would be clearly visible in a link budget listing with the result being that the designer would select a different component in order to close the link.

The FODAP user can use this aspect of SAM to work backwards through the link. That is, he could specify a desired link power margin and have SAM compute the required source power to achieve that goal.

2.5.2 System Bandwidth Calculation

The overall system bandwidth is an important parameter. The designer must be aware of it to ensure that his design goal has been met. This is particularly true in analog systems. The main concern is that selection of the wrong fiber can lead to bandwidth limitations. It is assumed that bandwidth concerns have been accounted for in the transmitter and receiver designs.

The calculation is performed by first taking the rms sum of the individual module risetimes:

$$t_{r_t} = \left(t_{r_{Tx}}^2 + t_{r_f}^2 + t_{r_{Rx}}^2 \right)^{1/2} \qquad (2.5.2\text{-}1)$$

where $t_{r_{Tx}}$ is the light output risetime from TAM, t_{r_f} is the "risetime" of the fiber from FAM and $t_{r_{Rx}}$ is receiver risetime as calculated in RAM. The overall system bandwidth $f_{3\ dB_t}$ is then calculated using the standard relation:

$$f_{3\ dB_t} = \frac{0.35}{t_{r_t}} \qquad (2.5.2\text{-}2)$$

LINK BUDGET

AVAILABLE POWER SOURCE	________	mW	________	dBm
INPUT COUPLING LOSS			________	dBm
FIBER LOSS			________	dBm
SPLICE LOSS			________	dBm
CONNECTOR LOSS			________	dBm
OUTPUT COUPLING LOSS			________	dBm
RECEIVER POWER	________	mW	________	dBm

TARGET BER

REQUIRED RECEIVER POWER	________	mW	________	dB
MARGIN			________	dB
SYSTEM BANDWIDTH			________	Hz

Figure 2.5.1-1. Sample Link Budget

SECTION 3.0

FODAP USERS' GUIDE

3.0 FODAP USERS' GUIDE

The Fiber Optics Design Aid Package (FODAP) is a comprehensive set of fortran coded computer routines which perform an analysis of an optical cable communications system. The program is design oriented and interactive or batch in mode. As a tool for the communication system designer FODAP reduces the time necessary to perform simple system analyses and can thereby achieve a higher degree of optimization and accuracy.

FODAP is structured in a highly modular format to facilitate update and expansion. The routines fall into six basic sections: the executive (EXEC), data manager (MANAGER), transmitter analysis module (TAM), fiber cable analysis module (FAM), receiver analysis module (RAM), and the system analysis module (SAM). See Figure 3.0. The EXEC interprets user instructions and sees that the necessary modules are called as needed. The MANAGER performs all the standard file management functions, such as record creation and deletion, and directs and controls all data input. TAM, FAM, and RAM perform the necessary computations for their respective parts of the communication system. SAM collects data from the other three analysis modules and computes total system parameters.

Additionally FODAP is capable of storing and updating optical system data on a temporary or permanent basis. This enables the user to build a large data file describing many different devices which he may then put together in any combination during a design session. During analysis the user may execute or perform the TAM, FAM, RAM, and SAM modules either individually or as part of an overall system. A user may also cause FODAP to automatically vary one or more parameters over a specified range to provide optimization data.

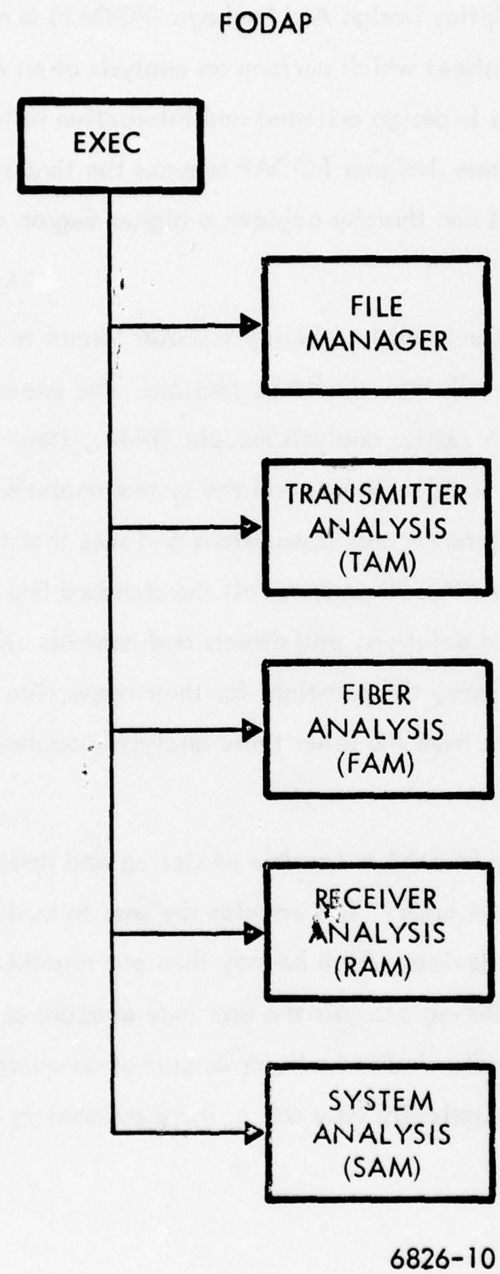


Figure 3.0. FODAP Organization

3.1 Guide Structure and Use

3.1.1 Organization

This users' guide has been organized to reflect the steps a designer must follow to use FODAP. Section 3.2 gives a detailed description of the data entry and management routines and their use. Sections 3.3 and 3.4 provide a general concept of how a single module analysis and a system (multimodule) analysis is performed. Sections 3.5 through 3.8 give detailed information and instructions on the use of each individual module. Section 3.9 describes several additional instructions recognized by FODAP which make the program easier to use.

3.1.2 Notational Conventions

The notational conventions used in this guide are listed below. The symbols *are not part of* the command line unless indicated in the text describing the statement.

<NAME>	Replace NAME with the appropriate parameter for the command line.
[NAME]	Name is an optional argument. If used replace with the appropriate parameter for the command line.
(NAME)	The name in parenthesis is a keyword and should be reproduced in the command line where indicated.
b̸	Insert one or more blank characters.

3.1.3 Command Format

All commands recognized by FODAP are free-form. There is no special column in which instructions must begin or end.

3.1.4 Name Conventions

All names used by FODAP, whether device record names or names of data items are limited to four characters. These characters may be any that are recognized by the host computer and the fortran compiler.

3.1.5 Definition of Terms

There are several basic terms used throughout this guide which should be defined. These are listed below.

device or component - A physical part of a fiber optics system, either a transmitter, a fiber cable, or a receiver.

module - A general reference to the portions of FODAP dealing in any way with a particular device. The transmitter module refers to the software analyzing the transmitter, the transmitter device records, and the transmitter data items. In addition, the segment of FODAP which deals with the calculation of the total system parameters is defined to be a module.

data item - One attribute or parameter of a device or component.

device record - A collection of data items which describe a device and are written on mass storage as a single record.

record name - The name given to a device record.

record address - A random access key assigned to a device record which gives the physical location of the record on mass storage.

module directory - An index to the device records of a module, consisting of the record names and associated record addresses.

module file - The set of device records and directory associated with a module.

data file - The single mass storage file which contains all the module directories and module files.

current data - The data items which have been read into FODAP and would be used for purposes of analysis if a module were executed. Each module has its own set of current data.

data name - The FORTRAN name associated with a data item.

3.2 Data Entry

3.2.1 Methods of Data Entry

Two methods used by FODAP for data entry are file input and item definition. In file input the MANAGER requests data from the user during the creation of a device record and that data is thereafter referenced by the assigned record name. This method is used for entering multiple data items on a system component or device. Item definition is a data entry method used primarily for data editing and redefinition or automatic incrementation of parameter values. The use of each of these methods is fully described in Sections 3.2.2 and 3.2.3.

3.2.2 Data Entry by File

3.2.2.1 Description of Data File

FODAP uses as storage for its device records a direct access file written in free format. The number of device records allowed per module depends on the size of the module directory record and is determined in the program source code. As shown in Figure 3.2.2.1, each module directory record is physically followed by its associated device records, referred to in the figure as data files. The location of the module directory records within the data file are also determined by the source code. Each device record entered requires two words of its module directory, one for storage of the record name and one for the record address. FODAP currently allows 100 device records per module.

3.2.2.2 File Utility Commands

3.2.2.2.1 Record and Directory Creation (CREATE)

Purpose:

Initializes the directory index and creates the device records.

Format:

CREATE ƀ <PAR-1> FILE ƀ [<PAR-2>]

FILE STRUCTURE

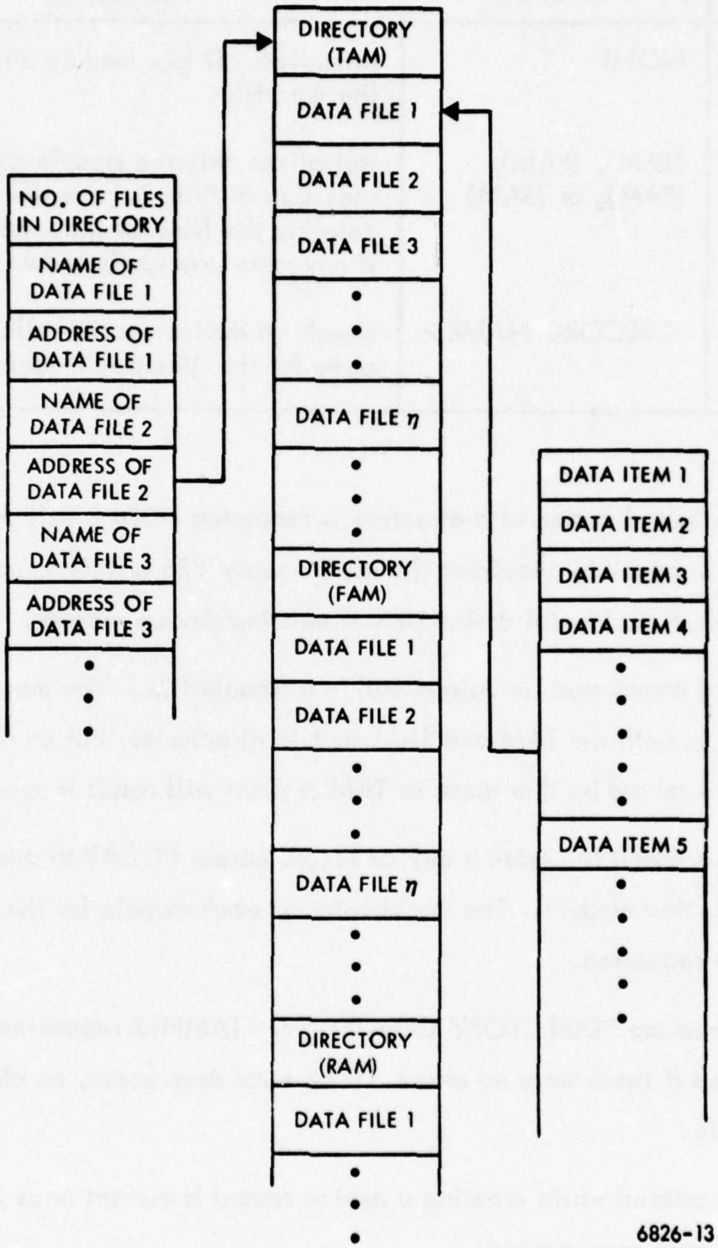


Figure 3.2.2.1. File Structure

Description:

<PAR-1>	<PAR-2>	Description
(NEW)	NONE	Initializes all four module directories in the data file.
(NEW)	(TAM), (FAM), (RAM), or (SAM)	Initializes only the module directory specified in PAR-2. Has the effect of deleting the files for the specified module if any were previously created.
(TAM), (FAM), (RAM), or (SAM)	<RECORD-NAME>	Creates a device record called record name for the specified module in PAR-1.

When initialization of a directory is requested FODAP will demand verification of the command by requiring the user to reply YES or NO to a query. This is done to protect against accidental destruction of existing device records.

Record names must be unique within a module file. You may have a device record named ONE in both the TAM and SAM module directories, but an attempt to create an additional device record by that name in TAM or SAM will result in an error.

The command to create a device record causes FODAP to ask for the appropriate data items for that module. See the chapter on each module for the exact data items which will be requested.

The message "DIRECTORY CREATED" or "TAMFILE record-name CREATED", etc., will be printed if there were no errors. If an error does occur, no changes will be made in the data file.

Data entered while creating a device record is current once FODAP has indicated no errors were encountered.

Examples:

1. CREATE NEWFILE

   The TAM, FAM, RAM, and SAM module directories are initialized. Any files which existed previously are effectively lost since their addresses are destroyed.

2. CREATE NEWFILE SAM

   Only the SAM module directory is initialized. Any previous SAM files are lost. The other module directories and device records are not affected.

3. CREATE TAMFILE ONE

   A transmitter device record called ONE is generated. The record name and its address is entered in the TAM module directory.

4. CREATE SAMFILE

   An error message is printed because a record name was not specified. FODAP waits for a new command.

3.2.2.2.2 Record Deletion (DELETE)

Purpose:

Deletes the specified device record from the module file.

Format:

DELETE ̸b <PAR> FILE ̸b <RECORD-NAME>

Description:

<PAR>(TAM) (FAM) (RAM) (SAM) - Specifies the module file where the device record to be deleted is stored.

<RECORD-NAME> - The name of the device record to be deleted.

The message "File Deleted" is printed when action is completed.

Example:

DELETE FAMFILE GE26

The device-record named GE26 is deleted from the FAM module directory.

3.2.2.2.3 Record Retrieval (GET)

Purpose:

Retrieve a device record which has been previously stored.

Format:

GET ƀ <PAR>FILE ƀ <RECORD-NAME>

Description:

<PAR> (TAM) (FAM) (RAM) (SAM) - Specifies the module file in which the device record is stored.

<RECORD-NAME> - The name of the device record to be retrieved.

A message such as "RAMFILE C13 RETRIEVED" is printed on successful retrieval.

The data items retrieved in the device record are current.

Example:

GET SAMFILE C

The device record C in the SAM module file will be read.

3.2.2.2.4 Record Update (SAVE)

Purpose:

To update a device record or to save data already read into the analysis program by writing it to the data file.

Format:

SAVE b̸ <PAR>FILE b̸ <RECORD-NAME>

Description:

<PAR> (TAM) (FAM) (RAM) (SAM) - Specifies the module file in which the device record is to be stored.

<RECORD-NAME> - The name of the device record to be updated or created.

The module data which is current is written to mass storage and stored under the record-name specified.

If a device record under the given record-name already exists, FODAP will demand verification of the command by requiring the user to reply YES or NO to a query. This is done to prevent accidental overwriting of a device record. If NO is returned, FODAP takes no action. If YES is returned, FODAP overwrites the contents of the earlier device record.

If a device record under record-name does not exist, FODAP creates a new device record and writes the data to mass storage.

Example:

SAVE TAMFILE X

All the current TAM data items would be written to the TAM module file under record-name X.

3.2.2.2.5 Record, Directory and Data Listing (LIST)

Purpose:

To list the contents of the device records, the current data in a module, or the record-names of device records in a module directory.

Formats:

1. LIST b̸ <PAR-1> FILE b̸ [<PAR-2>]

2. LIST b̸ <PAR-1> DIRECT

Description:

<PAR-1>	<PAR-2>	Description
(TAM), (FAM), (RAM), (SAM)	NONE	Format 1: Lists current data for the module specified by PAR-1.
(TAM), (FAM), (RAM), (SAM)	<RECORD-NAME>	Format 1: Reads and lists data stored in the specified module under record-name.
(TAM), (FAM), (RAM), (SAM)	NONE	Format 2: Lists the record-names of all device records stored in the specified module directory.

The data items are printed with their data names.

Examples:

1. LIST RAMDIRECT

   All record-names in the RAM module directory would be listed.

2. LIST FAMFILE

   All FAM data items current would be listed.

3. LIST SAMFILE T

   All SAM data items stored in the named device record would be read and then listed. Those data items read are current.

3.2.3 Item Entry

Although initially data items will be input to FODAP by file retrieval, it is necessary in a design-oriented program to be able to initially define or change the values of single variables. This is done by using the DEFINE statement.

Data items assigned values with the DEFINE statement do not regain their original values unless the data item is redefined or the appropriate device record it was stored in is retrieved again. If the user wishes to save the data items changed, a SAVE statement can be used.

3.2.3.1 Defining a Single-Valued Variable

Format 1:

DEFINE b̸ <DATA-NAME>=<VALUE>

Description:

<DATA-NAME> - The name of the data item to be defined. See the chapters on each module for a list of acceptable data names.

<VALUE> - The numeric value of the data item.

The value of the data item may be entered as a decimal number or in the fortran equivalent of scientific notation. Decimal points should always be included. Examples of acceptable numbers are:

1.2317	-66771.3	0.00014

Examples of numbers in scientific and E notation (ENTER NUMBERS ONLY IN E NOTATION):

-1.66×10^{-2}	-1.66 E-2
9.1×10^{23}	9.1 E23

3.2.3.2 Defining a Multivalued Variable

In many design problems a single variable must be varied over a range of values while holding other data constant. FODAP allows the user to vary these values automatically by using an alternate form of the DEFINE statement.

Format 2:

DEFINE ̸b <DATA-NAME>=<INITIAL-VALUE>,<FINAL VALUE>,
<INCREMENT>

Description:

<DATA-NAME> - The name of the data item to be defined. See the chapters on each module for a list of acceptable data names.

<INITIAL-VALUE> - The initial value of the data item.

<FINAL-VALUE> - The final value of the data item.

<INCREMENT> - The amount the data item is to be incremented for each analysis.

The values are entered in the same manner as described in Section 3.2.3.1.

An increment value of zero is not allowed.

Increment, initial, and final values may be positive or negative.

3.3 Analysis of a Module

3.3.1 Preparation for Module Execution

Before a module can be analyzed the user must be certain all the required data has been made available to the program. For a list of required data items see the chapter describing the appropriate module. To perform a module analysis use the appropriate command as follows.

Format:

TAM - transmitter analysis module

FAM - fiber cable analysis module

RAM - receiver analysis module

SAM - system analysis module

3.3.2 Execution Using Single-Valued Data Items

When all data items for the required module are single-valued one analysis is performed and an output table printed.

3.3.3 Execution Using Multivalued Data Items

When one or more data items for the required module are multivalued then multiple analyses are performed by FODAP and an output table is printed after each data item incrementation. For one multivalued data item the module is analyzed at the initial value of the item, the item is incremented, and the analysis performed with the new value. See Figure 3.3.3-1. The process is continued until the final-value is reached or the increment causes the data item to go beyond its defined range.

For more than one multivalued data item the module is first analyzed at the initial values of all the data items. The data items are then varied one at a time over their ranges until all combinations of all the values have been used in a module analysis. See Figure 3.3.3-2.

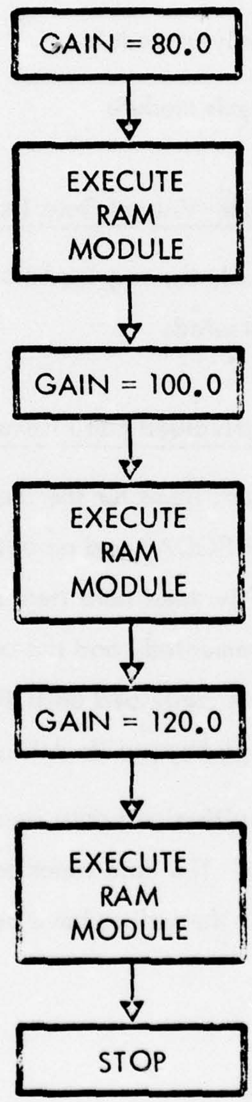
3.4 Analysis of a System

3.4.1 Preparation for System (Multimodule) Execution

The system designer is interested in seeing how various modules interact with one another to produce a working fiber optics communication system. This interaction is achieved by using the output values from one module as input values to the next module.

CONTROL CARDS: DEFINE GAIN = 80.0, 120.0, 20.0
RAM

EXECUTION SEQUENCE:

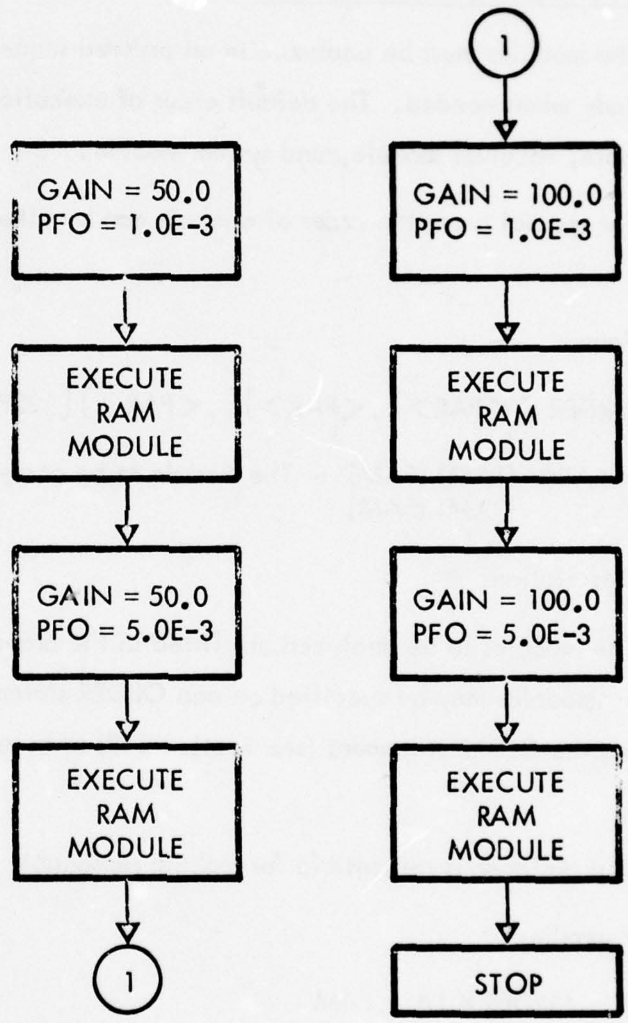


6826-35

Figure 3.3.3-1. Incrementation and Execution Sequence for a RAM Module With One Multivalued Input Variable

CONTROL CARDS: DEFINE GAIN = 50.0, 100.0, 50.0
DEFINE PFO = 1.0E-3, 5.0E-3, 4.0E-3
RAM

EXECUTION SEQUENCE:



6826-36

Figure 3.3.3-2. Incrementation and Execution Sequence for a RAM Module Analysis With Two Multivalued Input Variables

The data for each module must be already available to the program. For a list of the data items which are received from preceding modules see the appropriate sections on each module.

3.4.2 Ordering of Module Sequence

The modules must be analyzed in an ordered sequence to assure that all the input data is ready when needed. The default order of execution is the transmitter module, fiber cable module, receiver module, and system module.

For special cases the order of analysis can be altered using the ORDER statement.

Format:

ORDER b̸ <PAR>[,<PAR>][,<PAR>][,<PAR>]

<PAR> (TAM) (FAM) - The module to be analyzed.
(RAM) (SAM)

Description:

The modules to be analyzed are listed in the order they are to be executed. From one to four modules may be specified on one ORDER statement. The default order is restored by using the CLEAR statement (see Section 3.9) or by entering a new ORDER statement.

The designer is responsible for making reasonable use of this command.

Examples:

1. ORDER b̸ TAM,FAM

   The transmitter module, then the fiber cable module will be analyzed.

2. ORDER b̸ RAM,TAM,SAM

   The receiver module, the transmitter module, then the system module will be analyzed.

3.4.3 Execution of the Module Sequence

To perform an entire systems analysis, (the execution in sequence of the transmitter, fiber cable, receiver, and system modules) the SYS statement is used.

Format:

SYS

Description:

Each module will be executed in the specified order from the first to the last, unless altered by the use of the ORDER statement. The output values from each module are available as input to later modules which use them.

3.4.4 System Analysis With Single-Valued Data Items

When the input data items for all modules executed in a system analysis are single-valued, each module in the order sequence is executed once. Should an error occur during processing FODAP will cease execution and wait for a new command from the user.

3.4.5 System Analysis With Multivalued Data Items

One of the more powerful features of FODAP becomes apparent when a system analysis is performed on modules whose input data items are multivalued. This type of analysis requires that all combinations of values of all variables be used as input data to the system sequence. Many input variables however, are used in only one module and have no effect on the preceding modules executed in a sequence. FODAP takes advantage of this fact by using a "step-up" procedure after the initial pass through the sequence to avoid the execution of modules whose input values have not changed. Figure 3.4.5 illustrates this step-up by showing the value changes and order of module executions. The three variables which are defined as multivalued are PTI, the power input to the transmitter (a TAM data item); XL, the fiber cable length (a FAM data item); and XAD, the excess gain exponent (a RAM data item). At the start of the sequence each variable is set to its

CONTROL CARDS:

DEFINE PTI = 1.0E-3, 5.0E-3, 4.0E-3
DEFINE XL = 0.5, 1.0, 0.5
DEFINE XAD = 0.3, 0.4, 0.1
SYS

EXECUTION SEQUENCE:

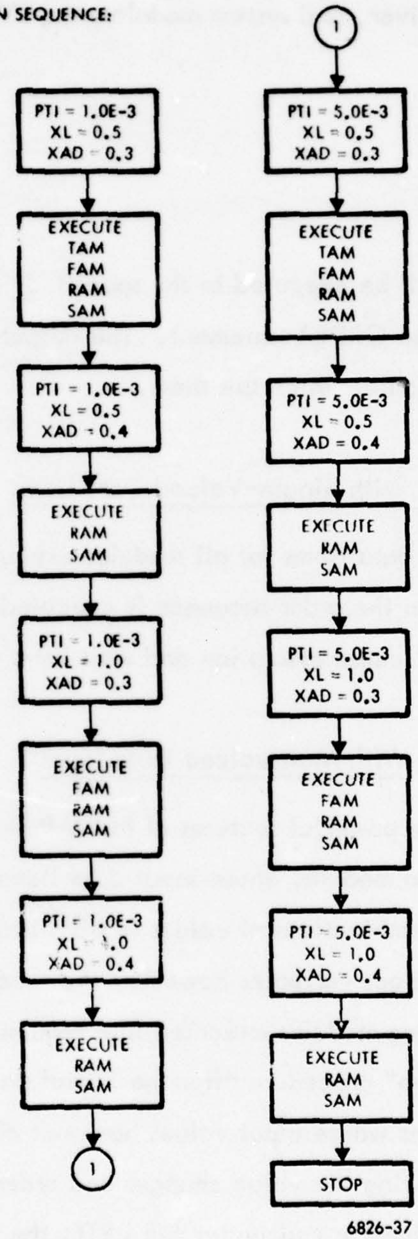


Figure 3.4.5. Execution Steps for System Analysis Using Multivalue Variables

initial value and all four modules are executed. Rather than incrementing the first variable next, FODAP chooses to increment XAD, the RAM variable, and execute only RAM and SAM. The TAM and FAM modules need not be executed again since their input data items were not changed. The FAM data item XL is incremented next, XAD is reset to its initial value, and FAM, RAM, and SAM are executed. This procedure is continued until all value combinations of the variables have been used as input. All variables are restored to their initial values at the conclusion of the analysis.

3.5 Transmitter Analysis Module

The transmitter analysis module is concerned with the parameters of only that part of a fiber cable system consisting of the transmitter and the fiber cable pigtail, if one exists. Since detailed descriptions of the four type of transmitters and the variables which describe them are found elsewhere in the FODAP document, this section will deal only with the actual computer execution of a transmitter analysis.

3.5.1 TAM Input Data

The TAM module, unlike FAM, RAM, and SAM, receives all input data through the data file or the DEFINE statement.

Input from the data file is requested by use of the GET TAMFILE NAME or LIST TAMFILE NAME commands. A GET TAMFILE NAME command (Section 3.2.2.2.3) causes FODAP to read the named device record from mass storage. A LIST TAMFILE NAME command (Section 3.2.2.2.5) causes FODAP to read the named device record from mass storage and also print the values.

Data is both read into FODAP and written to mass storage as a device record by using CREATE TAMFILE NAME. A CREATE TAMFILE NAME command (Section 3.2.2.2.1) causes FODAP to print a list of variables which the user should enter into the program. This directed entry of data is intended to reduce error by forcing a number to be input for every requested variable. These numbers are entered in the order asked for by the program, separated by commas, and in the format described in Section 3.2.3.1.

The necessary data items could also be entered individually with the DEFINE statement. The user assumes the responsibility of including every data item needed when using this entry method.

Table 3.5.1-1 lists all variables defined as input for TAM. The data names are the FORTRAN names recognized by FODAP and are always used as shown.* Notice that different sets of input variables are needed for the various source types shown. The designer should use this table as a checklist before attempting to enter his data into FODAP to be certain values exist for all necessary data items.

Table 3.5.1-2 is a list of coded input variables and the values those variables may be assigned. The designer should be sure to use only those values indicated.

3.5.2 Execution of the TAM Module

As described in Section 3.3.1 analysis of a transmitter is initiated by use of the command directive,

TAM

The TAM module does not attempt to read any data since it assumes the user has already entered the data through the use of the CREATE TAMFILE, GET TAMFILE, LIST TAMFILE, or series of DEFINE statements. TAM performs the task of calculating the parameters which describe the transmitter, printing the output table, and making available parameters which must be used by later modules.

It may be helpful to the designer to be aware of when and where certain parameters are computed. This information can be obtained by use of the provided logic diagrams (Section 3.12) and the program listings (Section 3.13) of FODAP. Notes are also provided with the module tables where helpful to describe more fully the nature of the variables listed.

---

*See Section 3.10 for a master list of all the variables and their definitions.

Table 3.5.1-1. Input Variables for TAM

	Edge LED		Surface LED		Laser (ILD)		Laser (YAG)	
	No Pigtail	Pigtail	No Pigtail	Pigtail	No Pigtail	Pigtail	No Pigtail	Pigtail
SRCE[1]	R	R	R	R	R	R	R	R
TAIL[1]	-	R	-	R	-	R	-	R
XLEN[1]	-	-	-	-	OPT	OPT	-	-
PTI[2]	R	R	R	R	R	R	R	R
XIO[2]	-	-	R	R	-	-	-	-
BTR[2]	-	-	R	R	-	-	-	-
RT[3]	R	R	R	R	R	R	R	R
BW[3]	R	R	R	R	R	R	R	R
DS[4]	-	-	-	OPT	-	-	-	-
ANGL	-	R	R	R	-	R	-	R
SPEC[5]	-	-	-	-	-	-	-	-
XLAM[5]	-	-	-	-	-	-	-	-
TNA	-	R	-	R	-	R	-	R
TFP	-	R	-	R	-	R	-	R
TDF	-	R	-	R	-	R	-	R

(R) - Required input (OPT) - Optional input (-) - Not applicable

[1] These are coded variables. Values are given in Table 3.5.1-2.

[2] Enter one of the three variables PTI, XIO, or BTR when SRCE is a surface LED. If more than one variable has a value TAM will use in order of preference PTI, XIO, then BTR.

[3] Enter RT or BW but not both.

[4] DS must be defined if BTR is used.

[5] SPEC and XLAM describe the transmitter but are not used by TAM. They are used by the FIBER module and should be defined if possible.

Table 3.5.1-2. Coded Input Variables and Values for TAM

Variable	Coded Value	Definition
SRCE	1	Edge LED
	2	Surface LED
	3	Laser (ILD)
	4	Laser (YAG)
XLEN	0	No Lens
	1	Lens
TAIL	0	No Pigtail
	1	Pigtail

3.5.3 TAM Output Data

The TAM module prints at the end of every transmitter analysis an output table which contains variables of interest, both input and calculated, to the system designer. See Table 3.5.3 for a listing of those variables.

Other values are passed through COMMON for use in later modules. SPEC and XLAM are two variables which are attributes of the transmitter yet used by FAM. PTO is calculated and printed by the TAM module, but is also passed to both FAM and SAM. The values passed to other modules in this manner are listed as input variables to those modules in their input tables.

Table 3.5.3. Variables Listed in TAM Output Table

Variables		
SRCE	TAIL	BW
PTI	PR	PTO
XIO	BTR	XLAM
SPEC	TNA	TFP
TDF	ANGL	XAI[1]

[1] XAI is computed in the TAM module only if there is a pigtail and XAI is not given a value either with a FAM device record or with a DEFINE statement.

3.6 Fiber Analysis Module

The fiber analysis module is concerned with the parameters of only that part of a fiber cable system consisting of the fiber cable itself. FODAP distinguishes between types of fiber cables only by the values the cable parameters assume. A complete discussion of the fiber cable can be found elsewhere in the FODAP document. This section deals only with the actual execution of a fiber cable analysis.

3.6.1 FAM Input Data

The FAM module can receive input data from three sources; the data file, the TAM module, and the DEFINE statement.

Input from the data file is received from two device records; one FAM record and one SAM record. These records can be requested by using the GET FAMFILE NAME, GET SAMFILE NAME, LIST FAMFILE NAME, and LIST SAMFILE NAME commands. Data can also be entered through use of the CREATE statement. The operations of these commands are identical to the description in Section 3.5.1 for the TAM module.

The necessary data items could also be entered individually with the DEFINE statement. As stated in Section 3.5.1, the user assumes the responsibility of including all data items which should be defined.

Data is also passed to the FAM module from the TAM module. This is done automatically and is transparent to the user. If the TAM module is not executed prior to the FAM module, the designer must be certain the data items which would have been defined are entered by another method.

Table 3.6.1 lists all variables defined as input for FAM. The data names are the FORTRAN names recognized by FODAP and are always used as shown. See Section 3.10 for a master list of all the variables and their definitions. The designer should use this table as a checklist before attempting to enter his data into FODAP to be certain values exist for all necessary data items.

Several variables are assigned default values as shown in Table 3.6.1. These values are used only if the listed parameters have a value of zero when the FAM analysis begins.

3.6.2 Execution of the FAM Module

As described in Section 3.3.1 analysis of a fiber cable is initiated by use of the command directive

FAM

The FAM module does not attempt to read any data items. The user is assumed to have already entered all the necessary input data (see Section 3.5.2). FAM performs the tasks of calculating the parameters which describe information flow through the fiber cable, printing the output table, and making available parameters used by the RAM and SAM modules.

Table 3.6.1. Required Input Variables for FAM

	Input From FAM File	Input From TAM	Input From SAM	Input Only With DEFINE
PTO		X		
PFI[1]				X
CORE[3]	X			
DF	X			
FP	X			
DEL[3]	X			
ALFA	X			
XLMC	X			
CLAD[3]	X			
XNA[3]	X			
PN1[2]	X			
PDD[2]	X			
PSEC[2]	X			
AF	X			
AS	X			
AC	X			
XAI	X			
SRCE		X		
TAIL		X		
XLEN		X		
DS		X		
ANGL		X		
SPEC		X		
XLAM		X		
XL			X	
CN			X	
SN			X	
XAO			X	

[1] PFI and PTO are equivalent if the transmitter has a pigtail. If PFI is the input power only XLAM and SPEC need be defined from the TAM module. If both variables are defined, PFI is used.

[2] The three variables PN1, PDD, and PSEC have default values derived from an optical fiber made by Corning with a titanium oxide doped core and fused silica cladding, which is similar to a germanium doped fiber. The values are: PN1 = -0.0145, PDD = -0.00085 and PSEC = 0.025.

[3] The four variables CORE, CLAD, DEL, and XNA are interrelated. If the combinations of variables listed are entered the missing values will be calculated by FAM. Those combinations are XNA and DEL, CORE and CLAD, and CORE and DEL. New values will not be computed for any defined variables.

3.6.3 FAM Output Data

The FAM module prints at the end of every fiber cable analysis an output table which contains variables of interest, both input and computed, to the system designer. See Table 3.6.3 for a listing of those variables.

Table 3.6.3. Variables Listed in FAM Output Table

Variables		
DTRA	DTER	DTOT
BL	XLC	PTO
PFI	PFO	XLAM
SPEC	ALFA	DEL
XNA	CORE	CLAD
XAI[1]	LOSS	

[1] XAI is computed in the FAM module only if there is no pigtail on the transmitter and XAI is not given a value either by a FAM device record or a DEFINE statement.

Other values are passed through COMMON for use in RAM and SAM. The passed values are listed as input variables in the two receiving modules.

3.7 RAM Analysis Module

The receiver analysis module is concerned with the parameters of only that part of a fiber cable system consisting of the receiver. The nine different receiver types are discussed in detail elsewhere in this document. This section deals with the actual execution of a receiver analysis.

3.7.1 RAM Input Data

The RAM module receives input from three sources; the data file, the FAM module, and the DEFINE statement.

Input from the data file is received from two device records; one RAM record and one SAM record. These records can be requested by using the GET RAMFILE NAME, GET SAMFILE NAME, LIST RAMFILE NAME, and LIST SAMFILE NAME commands. Data can also be entered through use of the CREATE statement. The operations of these commands are identical to the description in Section 3.5.1 for the TAM module.

The necessary data items could also be entered individually with the DEFINE statement. The user assumes the responsibility of including all data items which should be defined.

Data is also passed to the RAM module from the FAM module. As mentioned in Section 3.6.1 this is done automatically and is transparent to the user. If the FAM module is not executed prior to RAM, the designer must be certain the data items which would have been defined are entered by another method.

Table 3.7.1-1 lists the source of all variables defined as input for RAM. The data names are the FORTRAN names recognized by FODAP and are always used as shown. Section 3.10 contains a master list of all the variables and their definitions.

Table 3.7.1-2 lists all variables input to RAM by the receiver type which uses the data. The designer should use this table as a checklist before attempting to enter his data into FODAP to be certain values exist for all necessary data items.

Table 3.7.1-3 is a list of the coded input variables and the values those variables may be assigned.

Table 3.7.1-1. Source of Input Variables for RAM

	Input From RAM File	Input From FAM	Input From SAM
TYPE[1,2]	X		X
GAIN	X		
XAD	X		
XIL	X		
XIB	X		
XISQ[2]	X		X
RES	X		
XM	X		
SRMS[2]	X		X
RNB	X		
XMSC	X		
XMA[2]	X		X
BETA[2]	X		X
R	X		
OPT[1]			X
TBER			X
TSNR			X
CHAN[3]			X
PFO		X	

[1]These are coded variables. Values are given in Table 3.7.1-3.

[2]The five variables noted are input from the SAMFILE if CHAN is greater than one. There will be one set of variables for each channel.

[3]If CHAN is greater than one the user must be sure to enter the variables necessary for each type of modulation. These may all be input with the DEFINE statement.

Table 3.7.1-2. Input Variables by Receiver Type for RAM

	Analog Baseband	Analog AM	Analog FM	Analog DSB	Analog SSB	Digital FSK	Digital PSK	Digital Baseband (OOK)	Digital Baseband (BPPM)
TYPE[1]	R	R	R	R	R	R	R	R	R
GAIN[2]	OPT	OPT	OPT	OPT	OPT	R	R	R	R
XAD	R	R	R	R	R	R	R	R	R
XIL	R	R	R	R	R	R	R	R	R
XIB	R	R	R	R	R	R	R	R	R
XISQ	R	R	R	R	R	R	R	R	R
RES	R	R	R	R	R	R	R	R	R
XM	R	-	-	-	-	-	-	-	-
SRMS	R	R	R	R	R	-	-	-	-
RNB	R	R	R	R	R	R	R	-	-
XMSC	-	R	R	R	R	R	R	-	-
XMA	-	R	-	-	-	-	-	-	-
BETA	-	-	R	-	-	-	-	-	-
R	-	-	-	-	-	-	-	R	R
OPT[1]	OPT	OPT	OPT	OPT	OPT	-	-	-	-
TBER	-	-	-	-	-	OPT	OPT	OPT	OPT
TSNR	OPT	OPT	OPT	OPT	OPT	-	-	-	-
CHAN[3]	OPT	OPT	OPT	OPT	OPT	OPT	OPT	OPT	OPT
PFO	R	R	R	R	R	R	R	R	R

(R) - Required input data
(OPT) - Optional input data
(-) - Not applicable

[1] These are coded variables. Values are given in Table 3.7.1-3.
[2] If undefined, GAIN is assumed to 1.0.
[3] If undefined, CHAN is assumed to 1.0.

Table 3.7.1-3. Coded Input Variables and Values for RAM

Variable	Coded Value	Definition
TYPE	1.0	Analog Baseband
	2.0	Analog AM
	3.0	Analog FM
	4.0	Analog DSB
	5.0	Analog SSB
	6.0	Digital FSK
	7.0	Digital PSK
	8.0	Digital Baseband (OOK)
	9.0	Digital Baseband (BPPM)
OPT	0.0	Do not optimize gain
	1.0	Optimize gain

3.7.2 Execution of the RAM Module

As described in Section 3.3.1 analysis of a receiver is initiated by use of the command directive

RAM

The RAM module does not attempt to read any data. As explained in Section 3.5.2 the user is assumed to have already entered the input data. RAM performs the tasks of calculating the parameters which describe the receiver and the information received, printing of the output table, and making available parameters used by the SAM module.

3.7.3 RAM Output Data

The RAM module prints an output table at the end of every receiver analysis which contains input and output variables of interest to the system designer. Table 3.7.3 is a list of those output variables.

Other values are passed through common for use in the SAM module. These passed values are listed as input variables by SAM.

Table 3.7.3. Variables Listed in RAM Output Table

Variables		
TYPE	OPT	GAIN
BAND	PFO	SNR/BER[1]
XAD	XISQ	XIL
XIB	RES	RNB
TSNR/TBER[1]	DPR	

[1] The variable appropriate for the receiver type is printed.

3.8 System Analysis Module

Unlike the other three modules of FODAP the SAM module concerns itself with all parts of a fiber cable analysis system. A discussion of the variables and output listed by SAM is found elsewhere in this document. This section deals with the execution of a system analysis only.

3.8.1 SAM Input Data

The SAM module can receive data from five sources; the data file, the TAM, FAM, and RAM modules, and the define statement.

Input from the data file is received from a SAM device record. This record can be requested by using the GET SAMFILE NAME or LIST SAMFILE NAME commands. Data can also be entered through use of the CREATE statement. The operations of these commands are identical to the description in Section 3.5.1 for the TAM module.

The necessary data items can be entered individually with the DEFINE statement. As stated in Section 3.5.1, the user assumes the responsibility of including all data items which should be defined.

Data is also passed directly to the SAM module from the TAM, FAM, and RAM modules. This is done automatically and is transparent to the user. Because of the nature of the SAM module it is unlikely a user would wish to execute SAM without executing the other three modules first. However, should this case arise the designer must be certain the missing data items are entered by another method.

Table 3.8.1 gives all variables input to SAM listed by source. The data names are the FORTRAN names recognized by FODAP and are always used as shown. See Section 3.10 for a master list of all the variables and their definitions. The designer should use this table as a checklist before attempting to enter his data into FODAP to be certain values exist for all necessary data items.

3.8.1.1 Input Data for Multichannel Analysis

The special input variables describing multichannel modulations can be entered or altered by a variation of the DEFINE statement.

Format:

DEFINE <DATA-NAME>(<CHANNEL-NO>)=<VALUE>

Description:

<DATA-NAME>	The data name of one of the five multichannel variables.
<CHANNEL-NO>	The channel number being defined.
<VALUE>	The value of the variable for the given channel.

Table 3.8.1. Input Variables for SAM by Source

	Input From SAM File	Input From TAM	Input From FAM	Input From RAM
CN	OPT			
SN	OPT			
XL	R			
XAO	OPT			
OPT	R			
TSNR/TBER[1]				
CHAN	OPT			
TYPE[2]	OPT			
SRMS[2]	OPT			
XISQ[2]	OPT			
XMA[2]	OPT			
BETA[2]	OPT			
PTO[3]		OPT		
PFI[3]			OPT	
PFO[3]			OPT	
XAI		OPT	OPT	
FIBR			OPT	
SPLC			OPT	
CONN			OPT	
SNR/BER[1]				OPT
XMAR				OPT
BL			OPT	
BAND				OPT
RT		OPT		

(R) - Required input variable
(OPT) - Optional input variable

---

[1] Either TSNR and SNR or TBER and BER will be input depending on the receiver type in the RAM module.

[2] The set of five variables TYPE, SRMS, XISQ, XMA, and BETA are entered only if CHAN is greater than one.

[3] One of the three variables PTO, PFI, or PFO will be used in the link budget for available power source. The variables are listed by priority.

Examples:

1. DEFINE TYPE (1.0) = 1.0

   This defines the modulation type for channel one to be analog baseband.

2. DEFINE BETA (10.0) = 1.0 E6

   This defines the BETA variable of channel 10 to have a value of 1.0 E6.

3.8.2 Execution of the SAM Module

As described in Section 3.3.1 analysis of a system is initiated by use of the command directive

SAM

The SAM module as the others does not attempt to read any data since it assumes the user has already entered it. SAM performs a few simple computations, prints the output table called the link budget, and passes data from its input file to the other modules.

3.8.3 SAM Output Data

The SAM module prints at the end of every analysis an output table called the link budget. The items listed in this link budget can be found in Table 3.8.3.

Other values from the SAM input file are passed through COMMON for use in TAM, FAM, and RAM. The passed values are listed as input variables from SAM in the input sections for each module.

Table 3.8.3. Output Items in Link Budget

Heading	Data Name
Available Power Source	PTO, PFI, PFO[1]
Input Coupling Loss	XAI
Fiber Loss	FIBR
Splice Loss	SPLC
Connector Loss	CONN
Output Coupling Loss	XAO
Receiver Power	PFO
SNR[2]	SNR
Target SNR[2]	TSNR
BER[3]	BER
Target BER[3]	TBER
Required Receiver Power	DPR
Margin	XMAR
System Bandwidth	SBW

[1] The first nonzero variable is used.

[2] SNR and TSNR printed only with analog receivers.

[3] BER and TBER printed only with digital receivers.

3.9 Other FODAP Commands

Several commands used by FODAP are not concerned with either data flow or the execution of analysis modules. Their purpose is to make it easier for the designer to use the fiber optics package. Each command is listed below along with an explanation of its use.

3.9.1 INPUT Statement

The purpose of the INPUT statement is to allow the designer to enter commands in either interactive or batch mode or some combination of the two. Batch mode in this case means FODAP will neither request user responses nor give prompting messages. Any errors encountered will cause program termination. The user may change modes at any time and as often as desired. The default mode is interactive.

Format:

1. INPUT ␢ IACT

   Causes FODAP to enter interactive mode.

2. INPUT ␢ CARD

   Cause FODAP to enter batch mode.

All FODAP commands are mode independent. Data entry while creating new device records is more difficult in batch mode only because the user is not prompted to enter each variable. Table 3.9.1 shows the order in which values should be entered while in batch mode. All values for a module are entered at one time.

3.9.2 CLEAR Statement

The CLEAR statement is used to clear all input data items, all arrays and flags, and reset all default values in FODAP. Only the mode, batch or interactive, is not reset. It is recommended that the designer use the CLEAR command after each problem to avoid any confusion resulting from old data values. The CLEAR statement consists of the single word

CLEAR

Table 3.9.1. Order of Entry of Module Variables While in Batch Mode

TAM	FAM	RAM	SAM[1]	SAM[2]
SRCE	CORE	TYPE	CN	TYPE
TAIL	DF	GAIN	SN	SRMS
XLEN	FP	XAD	XL	XISQ
PTI	DEL	XIL	XAO	XMA
ANGL	ALFA	XIB	OPT	BETA
DS	XLMC	XISQ	TBER	
XIO	CLAD	RES	TSNR	
BTR	XNA	XM	CHAN	
RT	PN1	SRMS		
BW	PDD	RNB		
SPEC	PSEC	XMSC		
XLAM	AF	XMA		
TNA	AS	BETA		
TFP	AC	R		
TDF	XAI			

[1] Record 1 is defined for every SAM file.

[2] Record 2 is defined only when CHAN >1.0. The five variables must be redefined for each channel number. Each set of five is one record.

3.9.3 END Statement

The END statement causes FODAP to close any files it has been using and terminate execution. The END statement consists of the single word

END

3.10 Variable Names, Units, Definitions

Table 3.10 is a list of all input and output variables used or computed by FODAP. The FORTRAN data names are in alphabetical order. Given for each name is the module device record where stored if the item is an input variable, the modules which use the data item, the physical units assumed, and a definition or description. Also given is the symbol used to identify the parameter in the Design Handbook, Section 2.0, if a symbol was assigned.

Notice that some variables are listed as both input and computed variables. This usually means a value will be computed for the variable if it is otherwise set to zero. In all cases, if a variable is not equal to zero FODAP will not attempt to redefine it.

Some variables are also shown as both not being input by file and not being computed. These variables have special purposes in the modules where they are listed as being used. See the appropriate section for details. Enter values for these variables with a DEFINE statement.

Table 3.10. FODAP Variable Definitions

FORTRAN Name	Device File	Modules Where Computed	Modules Where Used	Units	Definition
AC	FAM	—	FAM	dB/conn	Connector insertion loss
AF	FAM	—	FAM	db/km	Fiber attenuation coefficient
ALFA	FAM	—	FAM	—	Index gradient parameter
ANGL	TAM	—	TAM FAM	degrees	LED intensity output null angle, θ_n.
AS	FAM	—	FAM	dB/splice	Splice insertion loss
BAND	—	RAM	RAM SAM	Hertz	Receiver bandwidth, $f_{3\ dB\ R_x}$
BER	—	RAM	RAM SAM	errors/bit	Bit error rate, BER
BETA	RAM SAM	— —	RAM	—	Peak deviation/highest mod frequency (FM mod index); stored in SAM device file for multichannel analysis.
BL	—	FAM	FAM SAM	Hertz	Fiber bandwidth
BTR	TAM	—	TAM	watts/sr-cm^2	LED radiance, B
BW	TAM	TAM	TAM SAM	Hertz	Transmitter bandwidth, $f_{3\ dB\ T_x}$
CHAN	SAM	—	RAM SAM	—	Number of channels
CLAD	FAM	FAM	FAM	—	Fiber cladding refractive index index, n_2
CN	SAM	—	FAM SAM	—	Number of connectors in the fiber cable

Table 3.10. FODAP Variable Definitions (Continued)

FORTRAN Name	Device File	Modules Where Computed	Modules Where Used	Units	Definition
CONN	—	FAM	FAM SAM	dB	Total connector power loss
CORE	FAM	FAM	FAM	—	Fiber core refractive index, n_1
DEL	FAM	FAM	FAM	—	Fractional index difference, Δ
DF	FAM	—	FAM	microns	Fiber cable diameter, d_f
DPR	—	RAM	RAM	milliwatts	Receiver input power for target BER/SNR.
DS	TAM	—	TAM FAM	microns	Source diameter, d_s
DTER	—	FAM	FAM	nanoseconds	Intermodal dispersion
DTOT	—	FAM	FAM	nanoseconds	Total dispersion, σ_{total}
DTRA	—	FAM	FAM	nanoseconds	Intramodal dispersion, σ_{intra}
FIBR	—	FAM	FAM	dB	Total fiber attenuation
FP	FAM	—	FAM	—	Packing fraction, f_p
GAIN	RAM	RAM	RAM	—	Avalanche Gain
GOPT	—	RAM	RAM	—	Optimized gain request, G
LOSS	—	FAM	FAM	dB	Total loss due to connectors, splices, output coupling, and fiber attenuation.
OPT	SAM	—	RAM	—	Indicator variable for gain optimization.
PDD	FAM	FAM	FAM	—	$\lambda d\Delta / d\lambda$
PFI	—	—	FAM	watts	Power input to the fiber cable.

Table 3.10. FODAP Variable Definitions (Continued)

FORTRAN Name	Device File	Modules Where Computed	Modules Where Used	Units	Definition
PFO	—	FAM	FAM RAM	watts	Power at receiver end of the fiber cable, P_R
PN1	FAM	FAM	FAM	—	$\lambda dn_1/d\lambda$
PR	—	TAM	TAM	watts	Global power from transmitter. Without pigtail, power available to fiber cable. With pigtail, power before pigtail connection P_s.
PSEC	FAM	FAM	FAM	—	$\lambda^2 d^2 n_1/d\lambda^2$
PTI	TAM	—	TAM SAM	watts	Input power of transmitter, P_A
PTO	—	TAM	TAM FAM	watts	Power output from transmitter and available to fiber cable, P_A
R	RAM	—	RAM	bits/sec	Bit rate, R
RES	RAM	—	RAM	amp/watt	Responsivity of photodiode, $\sqrt{\ }$
RNB	RAM	—	RAM	Hertz	Noise equivalent bandwidth of receiver, b
RT	TAM	TAM	TAM	seconds	Rise time, t_r
SN	SAM	SAM	SAM	—	Number of splices in fiber cable
SNR	—	RAM	RAM SAM	—	Signal/noise ratio, SNR
SPEC	TAM	—	FAM	nanometers	Rms spectral width of source, σ_λ
SPLC	—	FAM	FAM SAM	dB	Total splice power loss

Table 3.10. FODAP Variable Definitions (Continued)

FORTRAN Name	Device File	Modules Where Computed	Modules Where Used	Units	Definition
SRCE	TAM	—	TAM FAM	—	Indicator variable for transmitter type
SRMS	RAM SAM	— —	RAM		Rms value of signal assuming $/s/_{max} = 1$; stored in SAM device file for multichannel analysis, $\sqrt{<s^2>}$
TAIL	TAM	—	TAM FAM	—	Indicator variable for transmitter pigtail
TBER	SAM	—	RAM SAM	errors/bit	Target or requested bit error rate
TDF	TAM	—	TAM	microns	Diameter of the transmitter pigtail, d_f
TFP	TAM	—	TAM	—	Packing fraction of pigtail, f_p
TNA	TAM	—	TAM	—	Numerical aperture of pigtail, NA
TSNR	SAM	—	RAM SAM	—	Target or requested signal/noise ratio
TYPE	RAM SAM	—	RAM	—	Indicator variable for receiver type; stored in SAM device file for multichannel analysis.
XAD	RAM	—	RAM	—	Diode constant (excess gain exponent), α_D
XAI	FAM	TAM FAM	TAM FAM	dBm	Input coupling loss
XAO	SAM	—	SAM	dBm	Output coupling loss
XIB	RAM	—	RAM	nanoamps	Bulk leakage current of photodiode, I_b

AD-A040 772 HARRIS CORP MELBOURNE FLA ELECTRONIC SYSTEMS DIV F/G 20/6
FIBER OPTICS DESIGN AID PACKAGE.(U)
MAY 77 I B SLAYTON, D J ESTAPA, J R JONES F30602-76-C-0246

UNCLASSIFIED RADC-TR-77-163 NL

2 OF 3
AD
A040772

Table 3.10. FODAP Variable Definitions (Continued)

FORTRAN Name	Device File	Modules Where Computed	Modules Where Used	Units	Definition
XIL	RAM	—	RAM	nanoamps	Surface leakage current of photodiode, I_L
XIO	TAM	—	TAM	watts/sr	Source intensity, I_o
XISQ	RAM SAM	—	RAM	nanoamps	Input referred preamp rms mean-square noise current; stored in SAM device file for multichannel analysis, I_A
XL	SAM	—	FAM	km	Length of fiber cable, L
XLAM	TAM	—	FAM	nanometers	Source emission wavelength, λ
XLC	—	FAM	FAM	km	Mode coupling length, L_C
XLEN	TAM	—	TAM FAM	—	Indicator variable for lensed transmitters
XLMC	FAM	—	FAM	dB/km	Excess loss induced by mode coupling, ℓ_e
XM	RAM	—	RAM	—	Peak IM index, M
XMA	RAM SAM	—	RAM	—	Mod index of AM waveform; store in SAM device file for multichannel analysis, M_a
XMAR	—	RAM	RAM SAM	—	Difference between the power available and the power necessary to achieve a target SNR or BER.
XMSC	RAM	—	RAM	—	Subcarrier mod index (0-1) optical mod index weighting relative to 100% intensity modulation, M_{sc}
XNA	TAM	—	TAM	—	Numerical aperture of fiber cable, NA

3.11 Assignment of Physical Units

Table 3.11 lists the logical files and FORTRAN unit numbers referenced by FODAP.

Table 3.11. File Assignments

FORTRAN Unit Number	Logical File Used
5	Command Input (interactive mode)
6	List Output
7	Command Input (batch mode)
17	Data File

The data file must be assigned to a mass storage device which allows random access I/O.

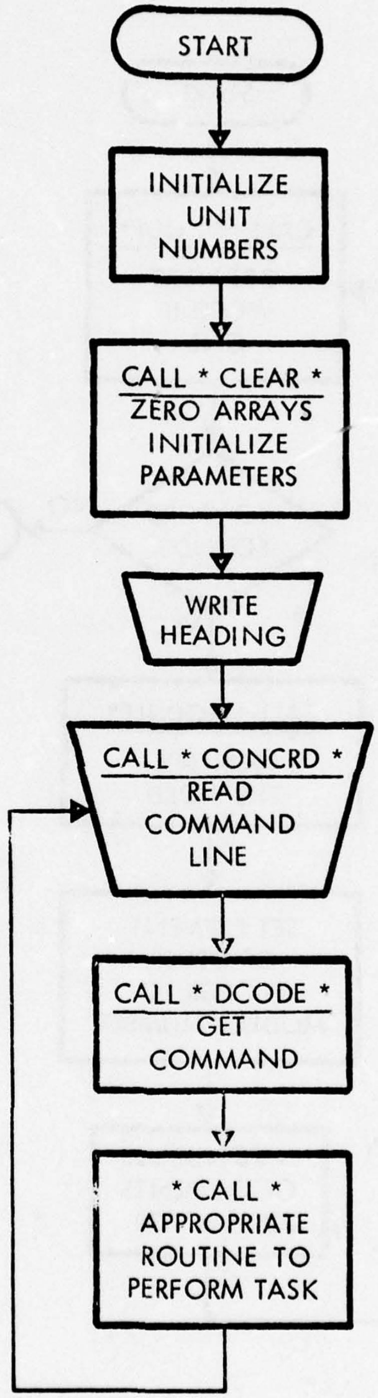
The user is responsible for assigning the listed FORTRAN unit numbers to the appropriate physical devices.

The device assignments for Command Input (interactive) and List Output on the Honeywell 6180 under Multics are default. The user need not assign these unit numbers before using FODAP.

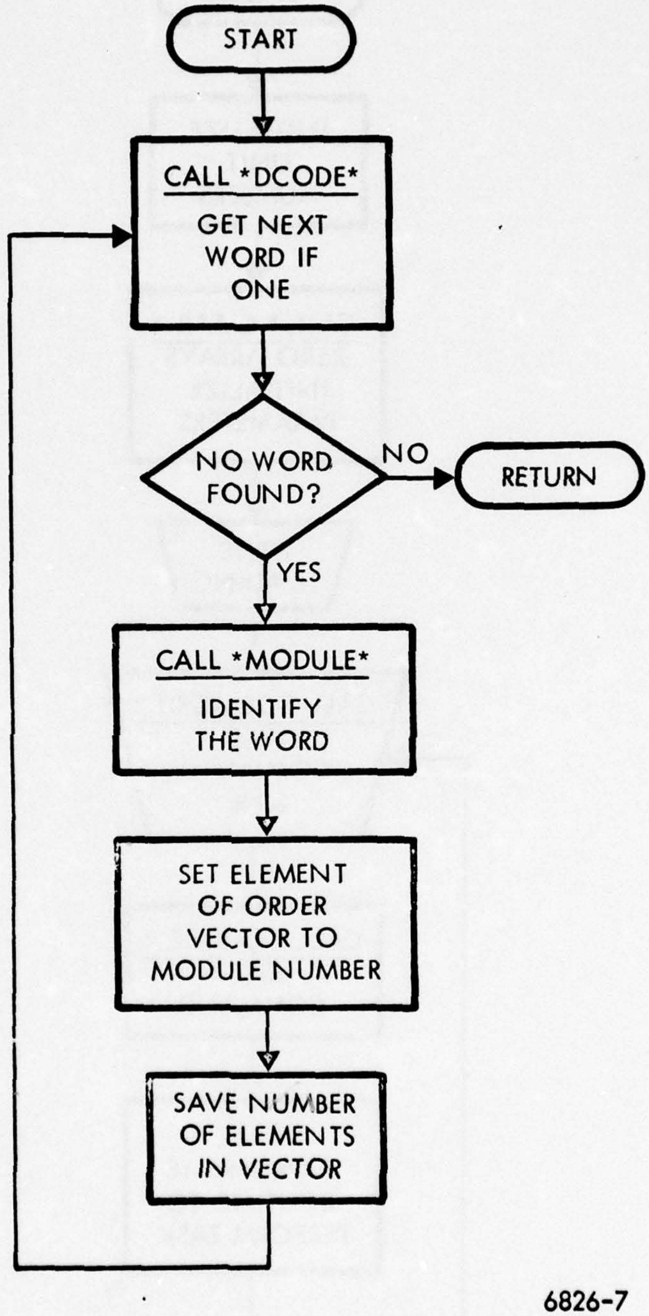
Assignment of a data file to unit number 17 before running FODAP is required. Remember to specify a keyed read/write file.

3.12 Logic Diagrams

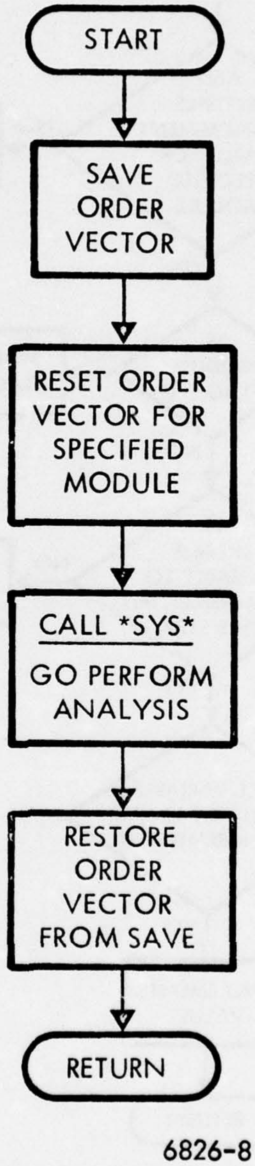
The logic diagrams on the succeeding pages are not intended to be detailed flowcharts of every FODAP routine. Instead it is hoped the diagrams will allow a user to understand in a general way what is happening within the program. To the programmer the diagrams should serve as a road map into the actual program listings, without going into so great a detail as to obscure the basic logic of the routines.



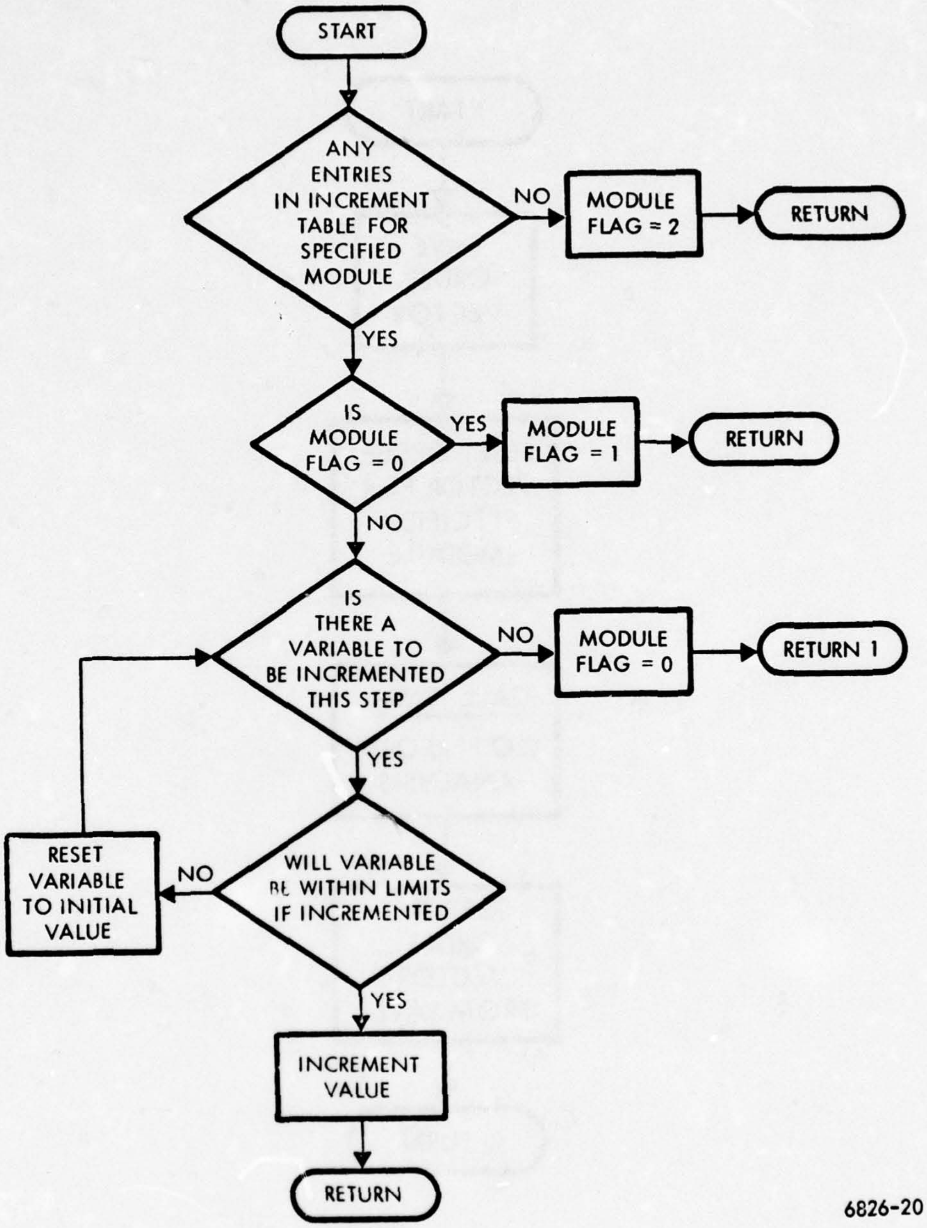
Logic Diagram for EXEC



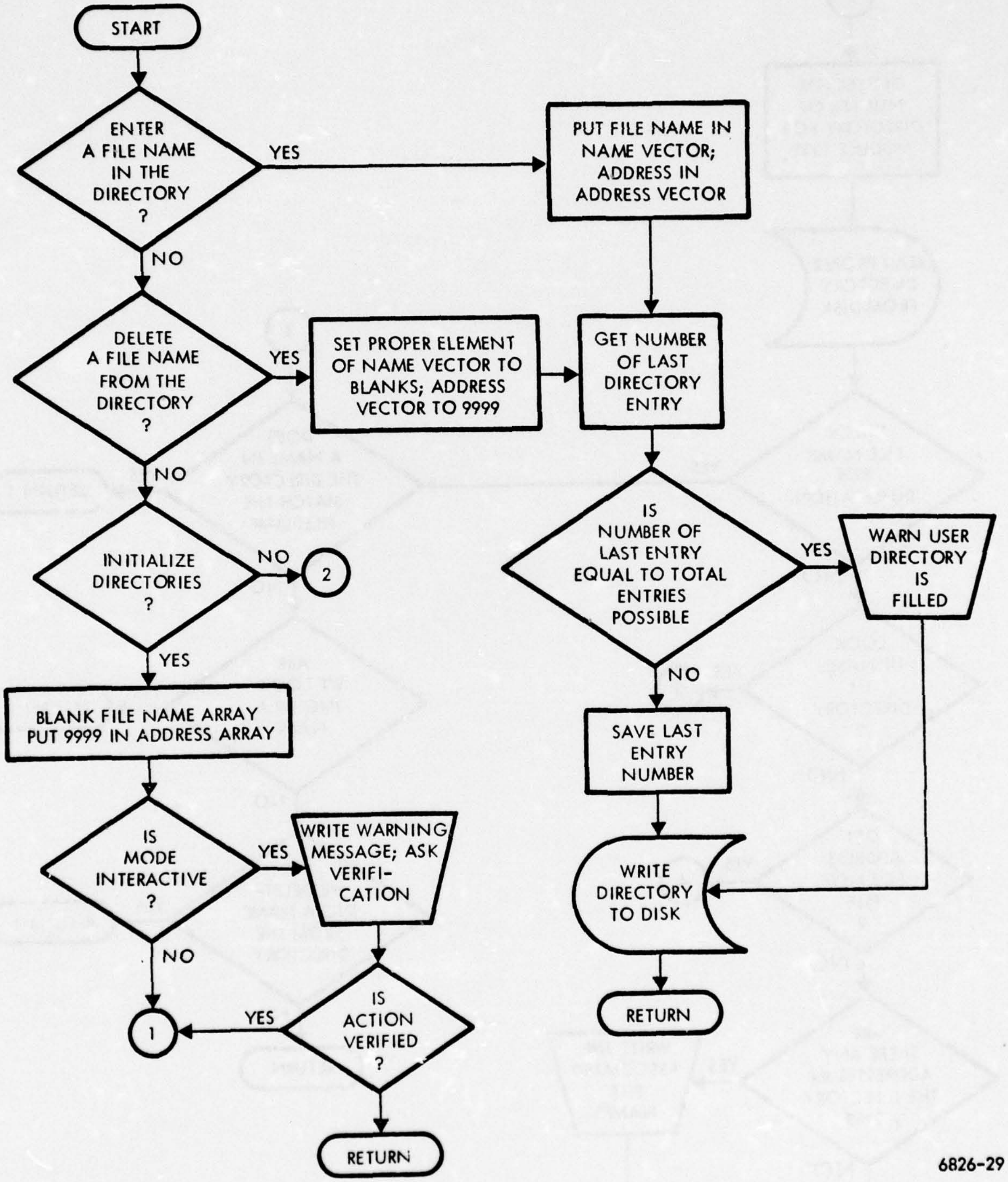
Logic Diagram for ORDER



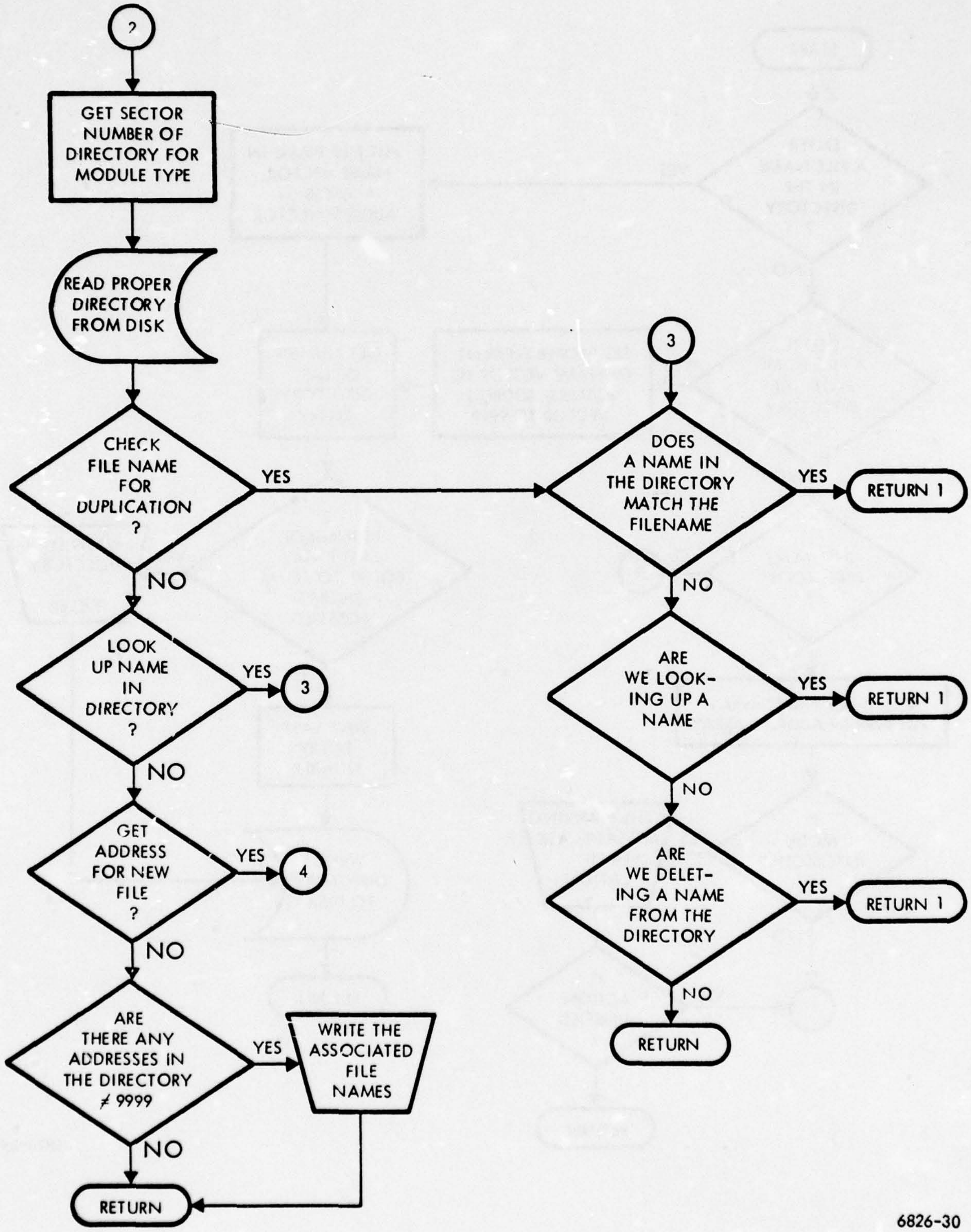
Logic Diagram for SET



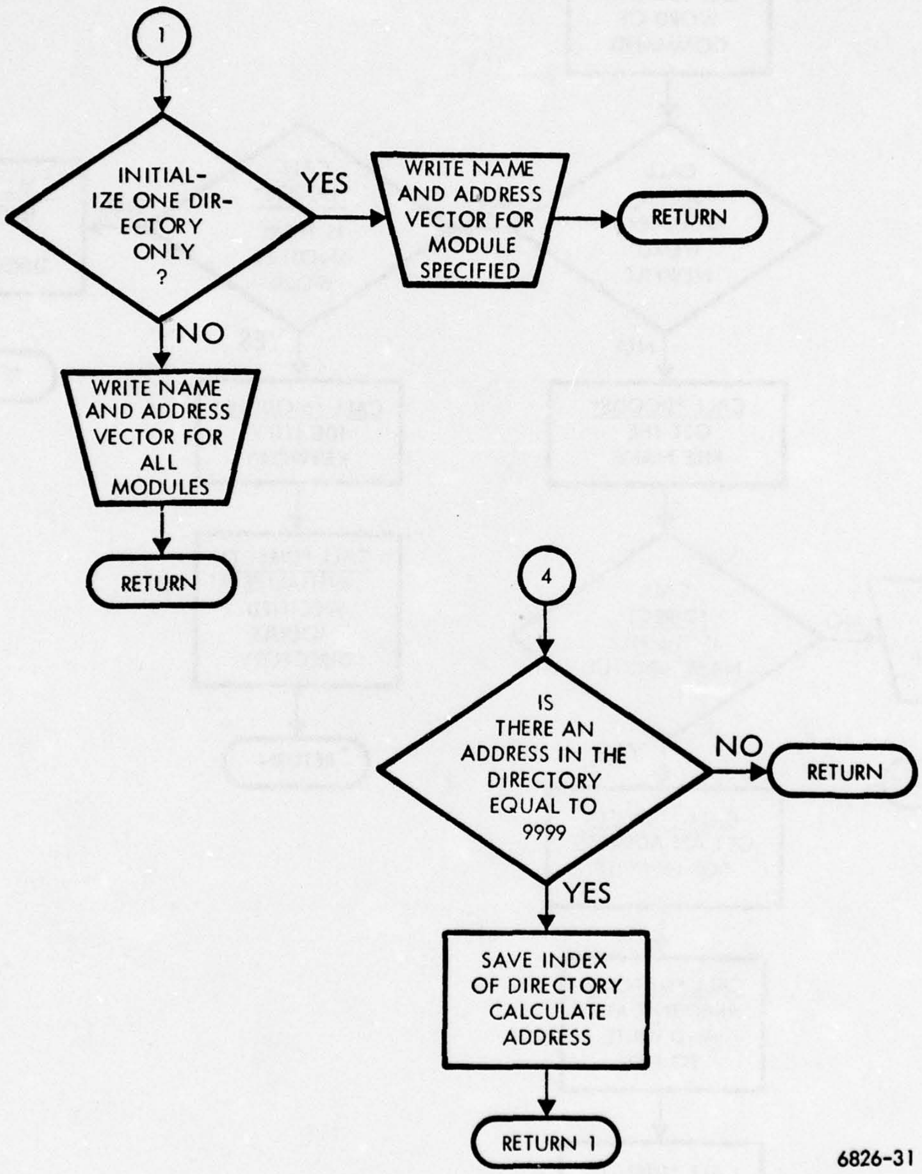
Logic Diagram for ALLOT



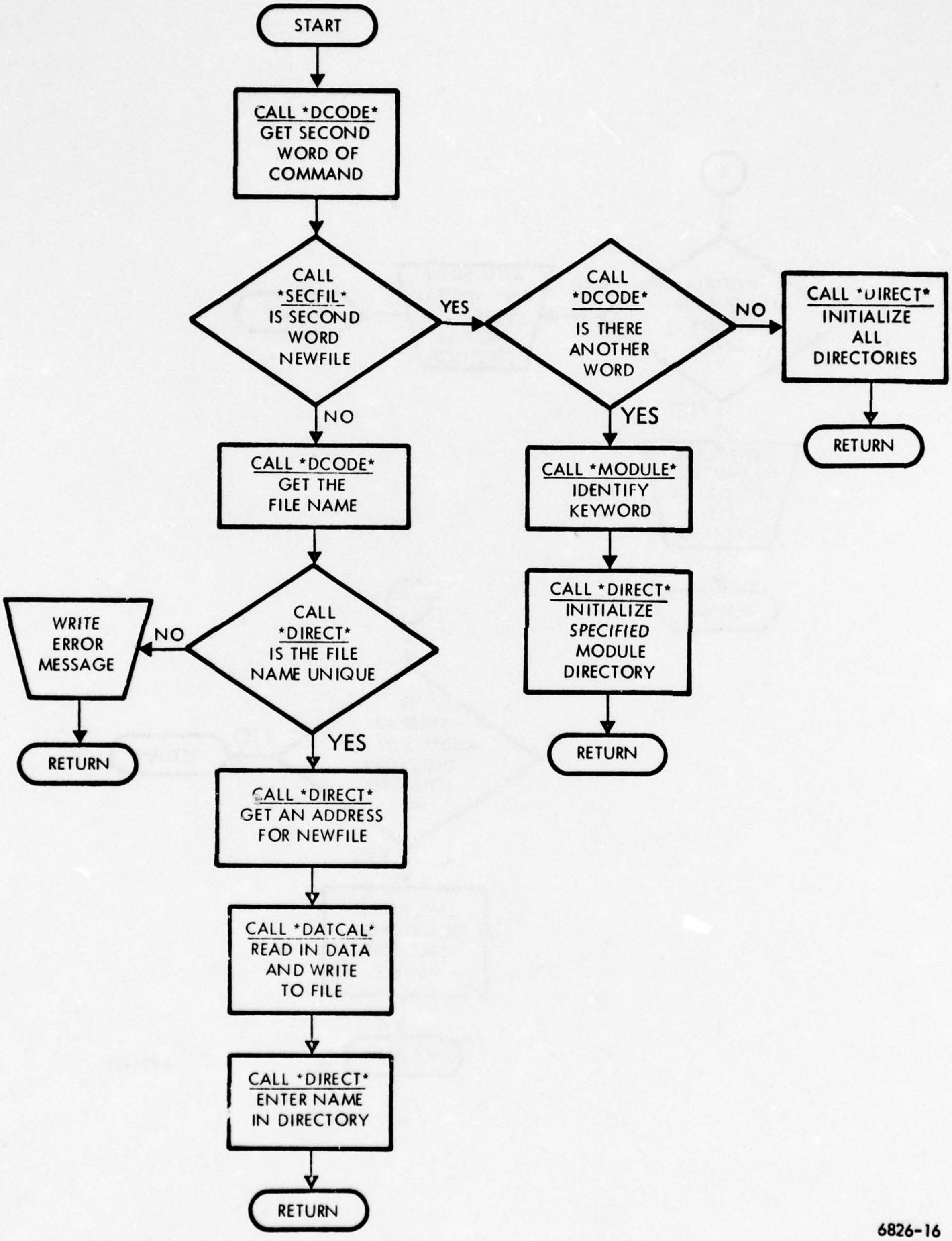
Logic Diagram for DIRECT (Sheet 1 of 3)



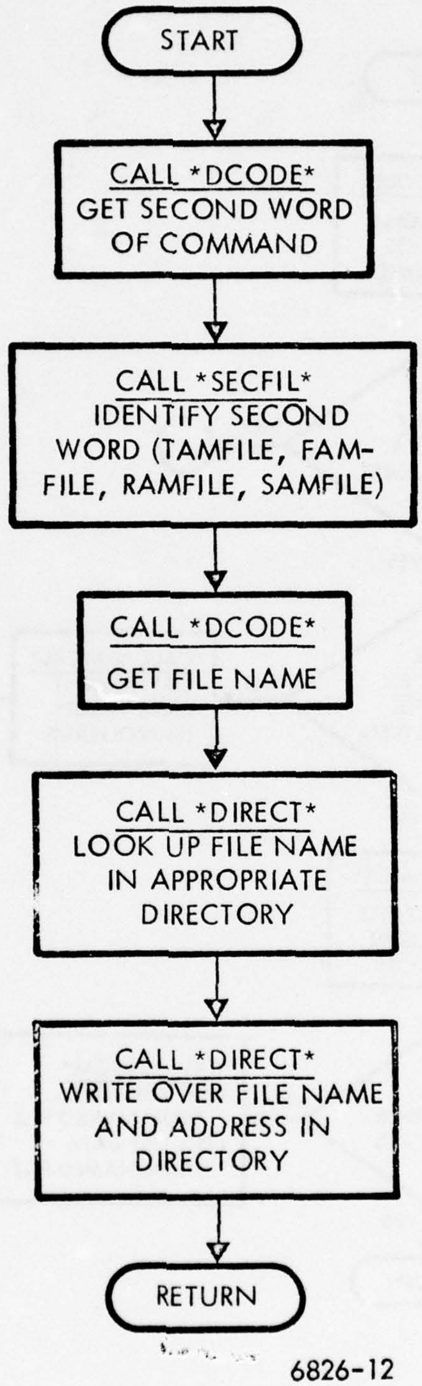
Logic Diagram for DIRECT (Sheet 2 of 3)



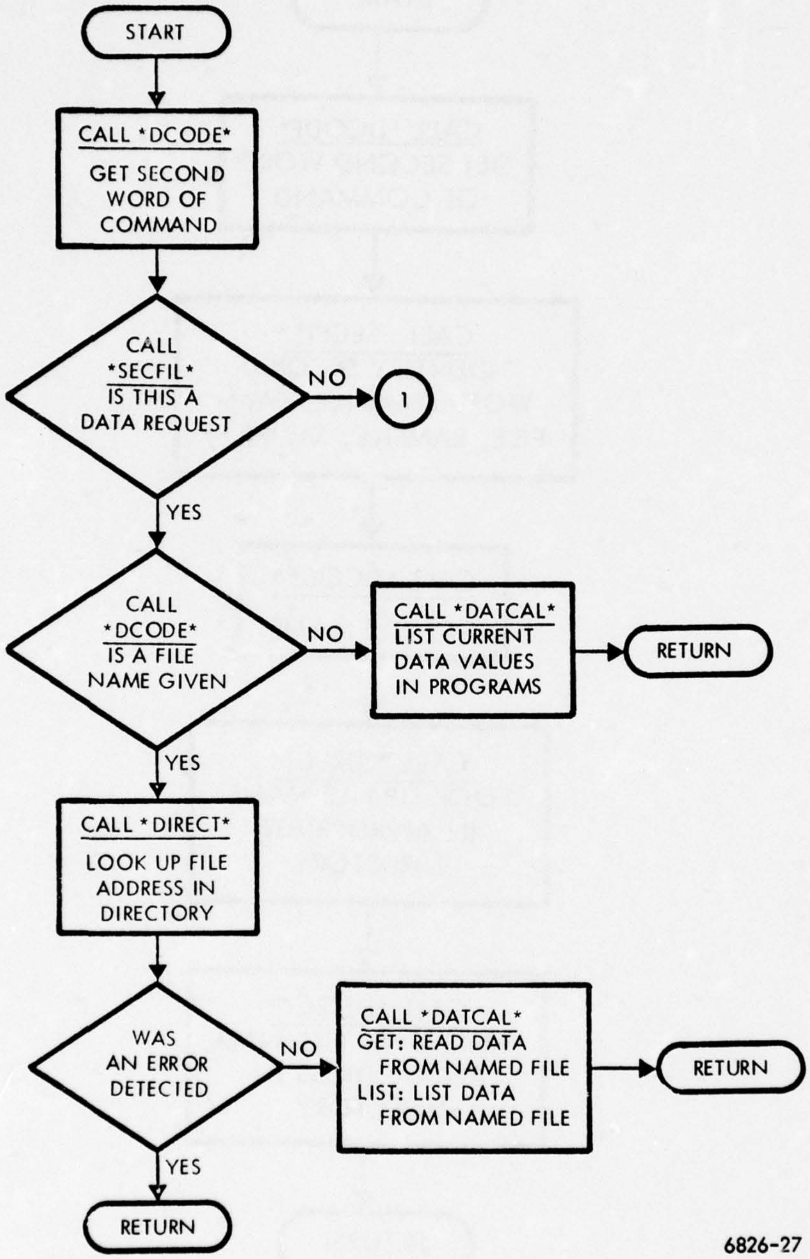
Logic Diagram for DIRECT (Sheet 3 of 3)



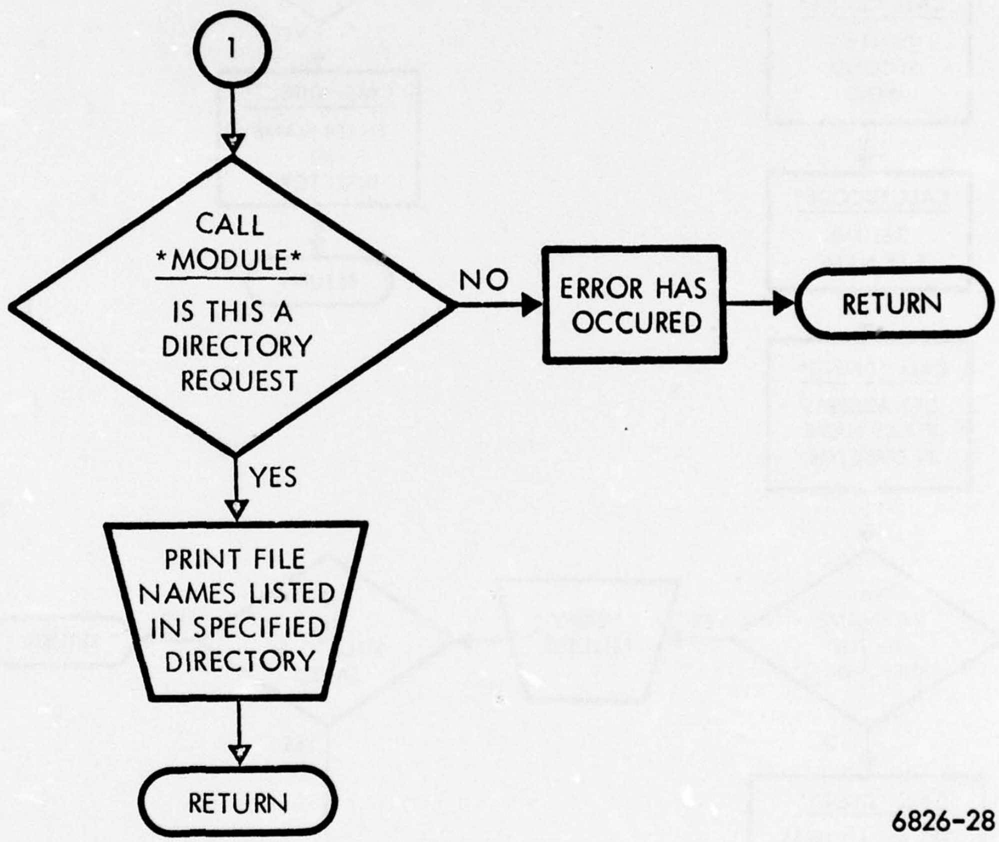
Logic Diagram for CREATE



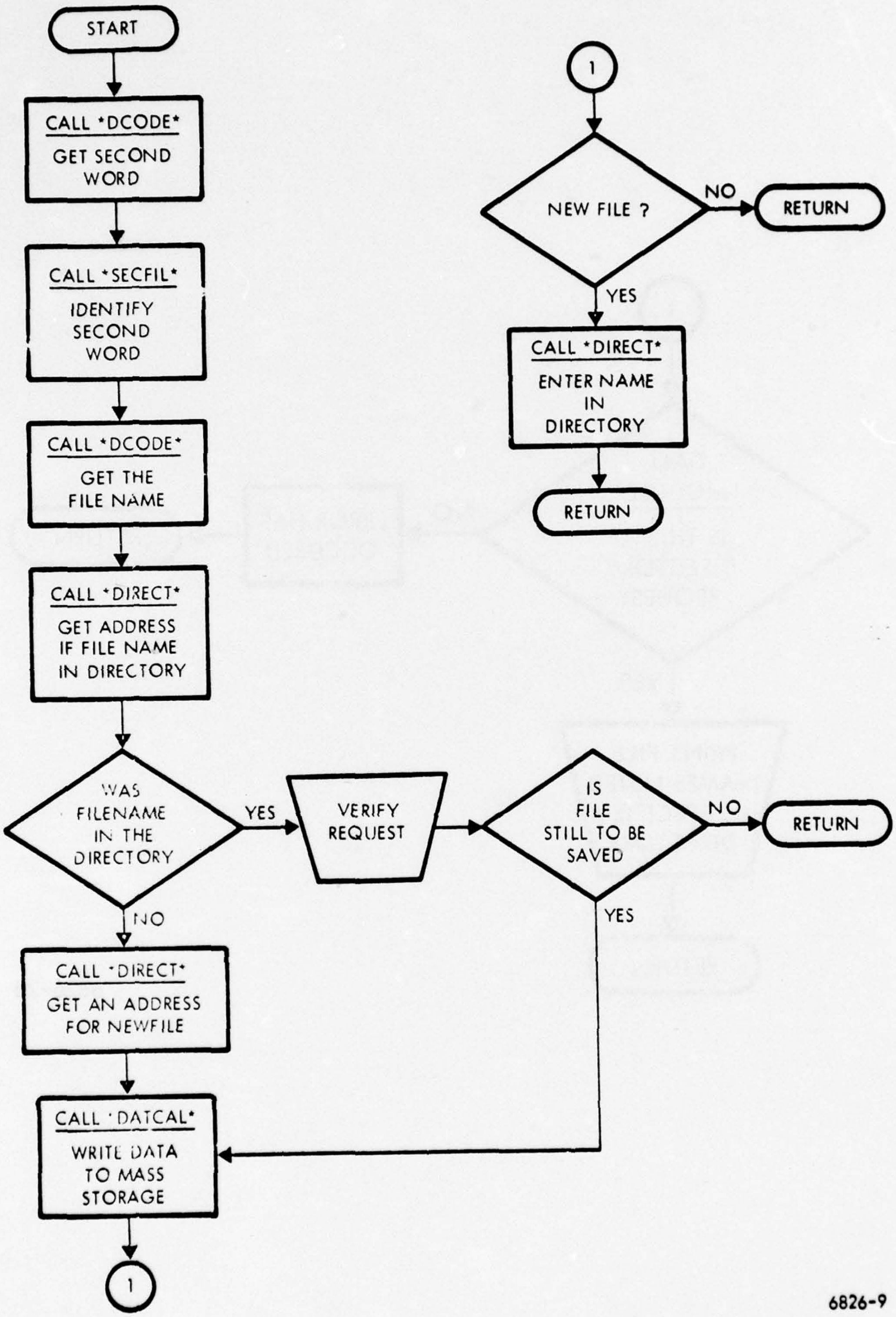
Logic Diagram for DELETE



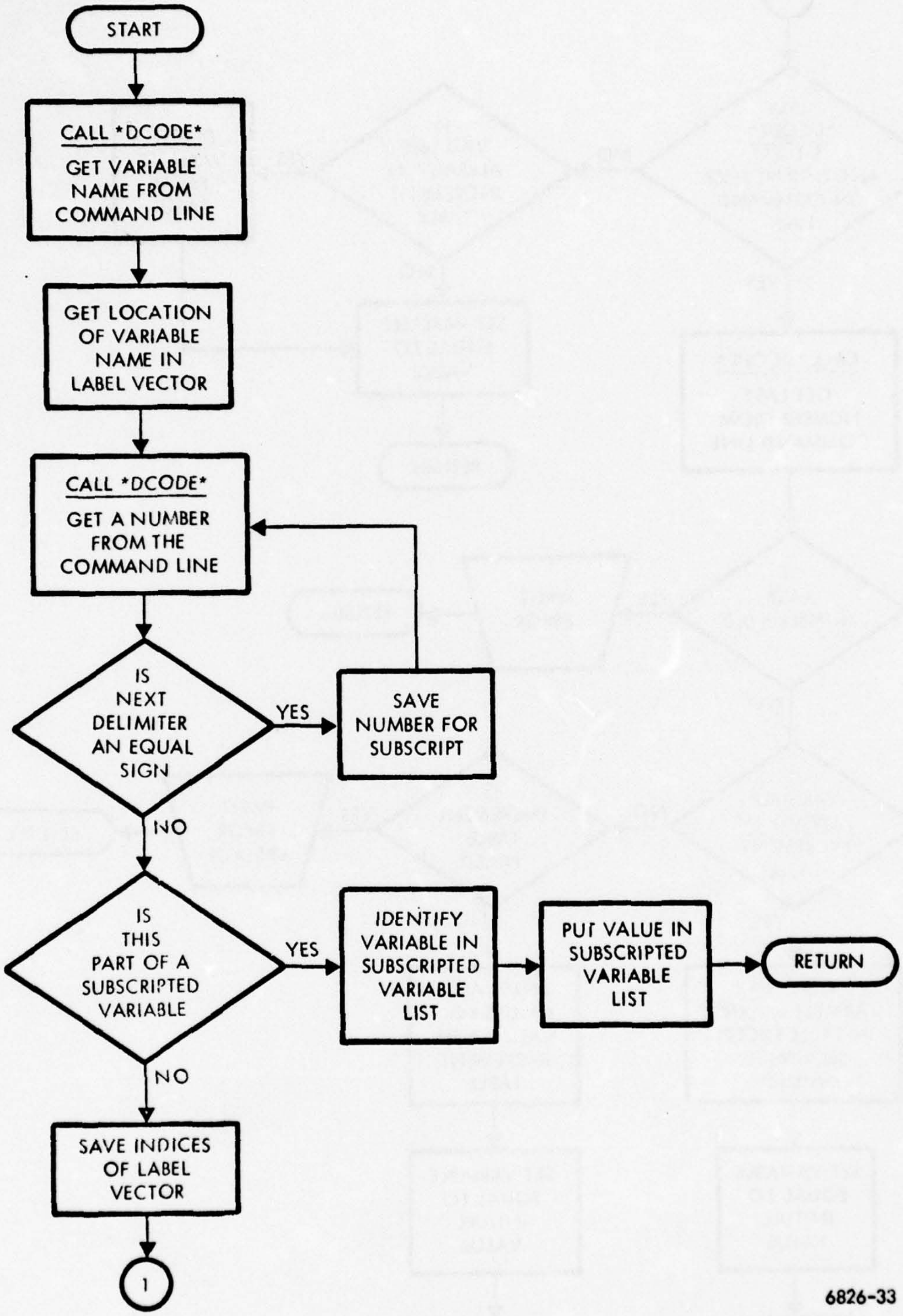
Logic Diagram for GET, LIST (Sheet 1 of 2)



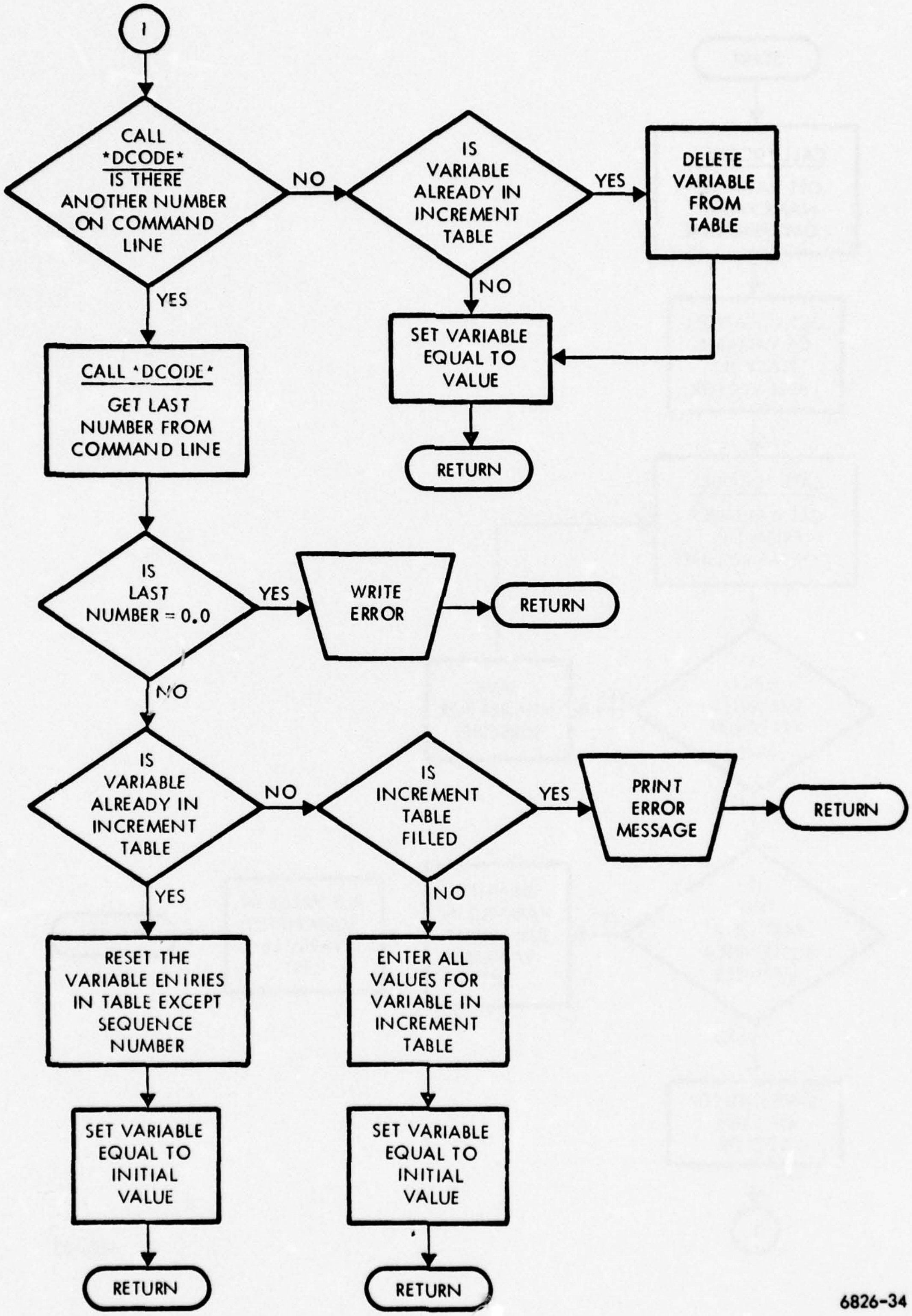
Logic Diagram for GET, LIST (Sheet 2 of 2)



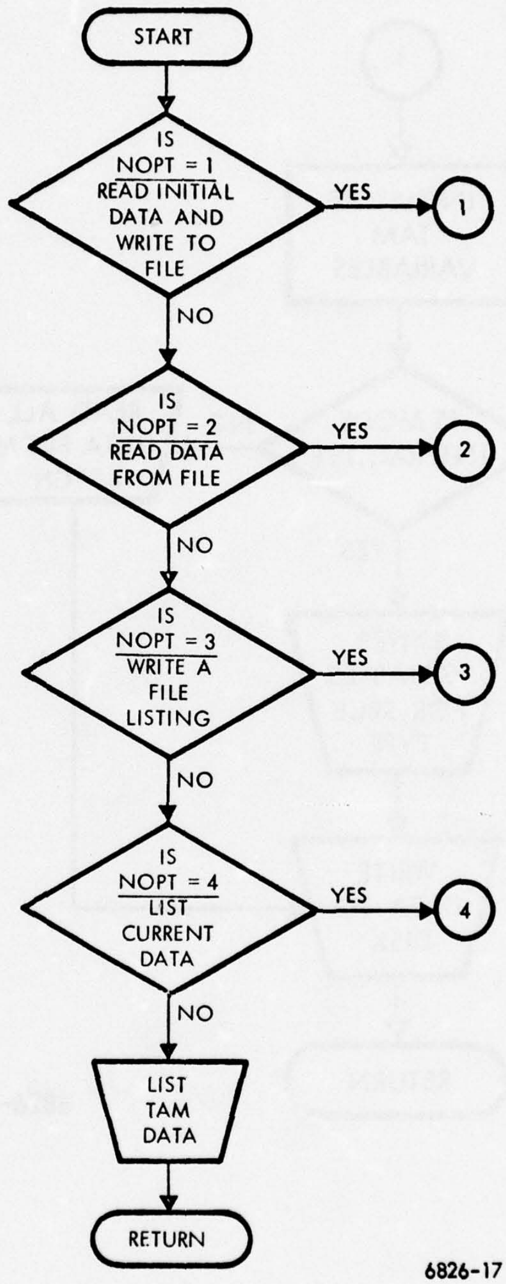
Logic Diagram for SAVE



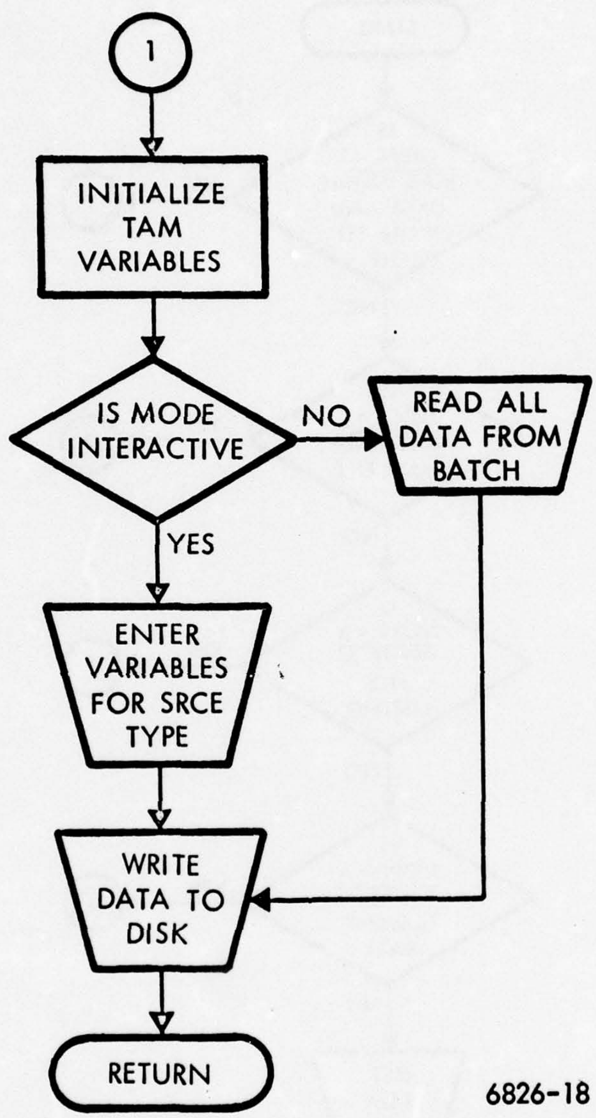
Logic Diagram for DEFIN (Sheet 1 of 2)



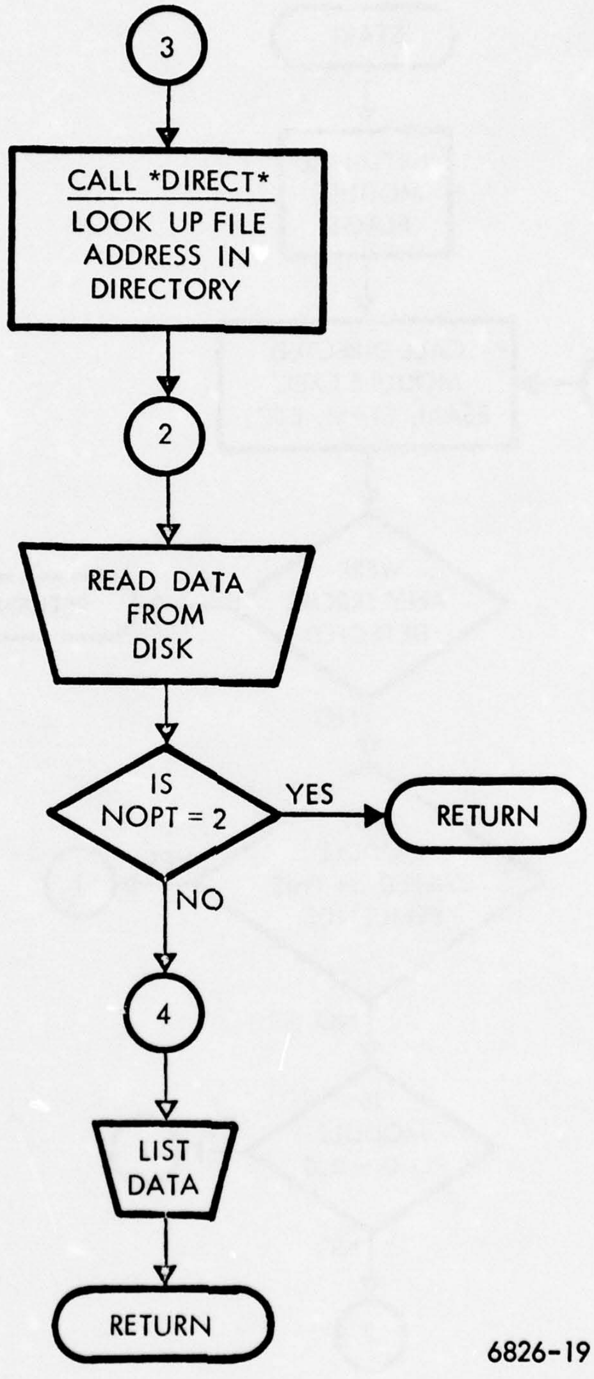
Logic Diagram for DEFIN (Sheet 2 of 2)



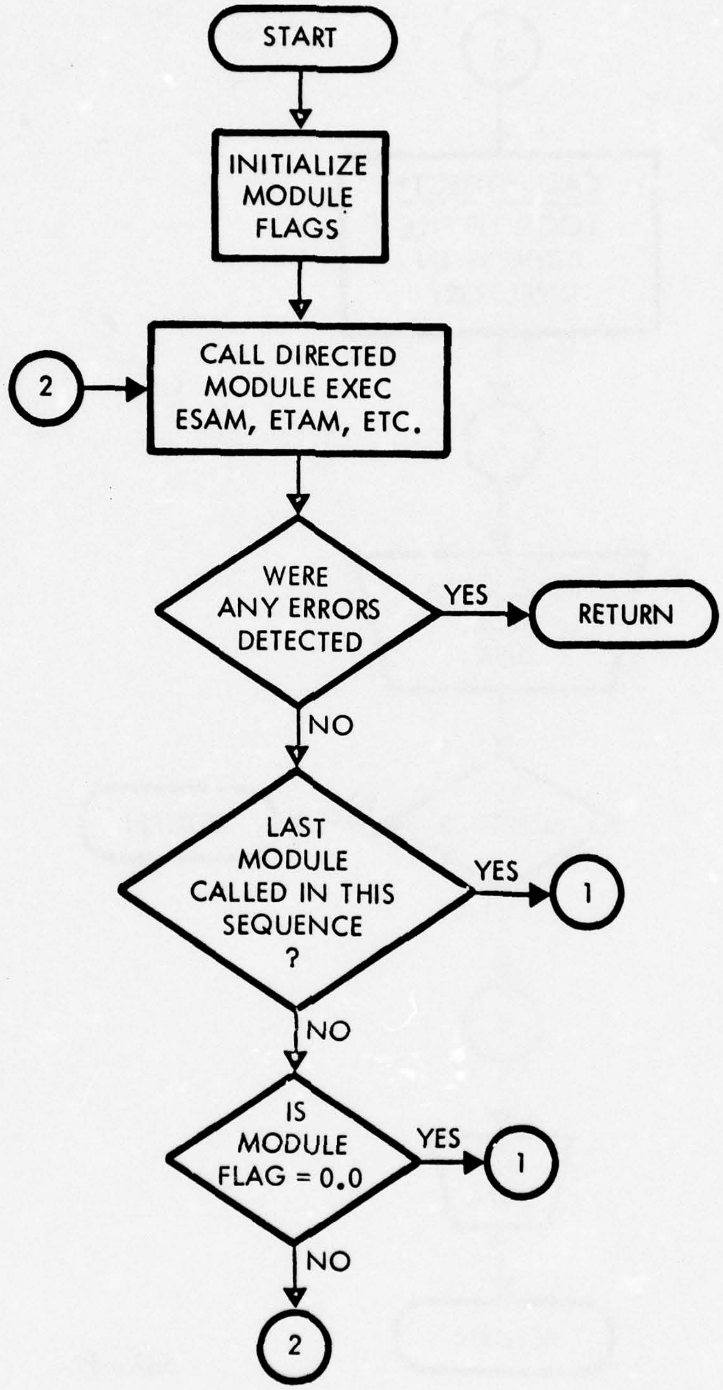
Logic Diagram for TAMDAT, FAMDAT, RAMDAT, and SAMDAT (Sheet 1 of 3)



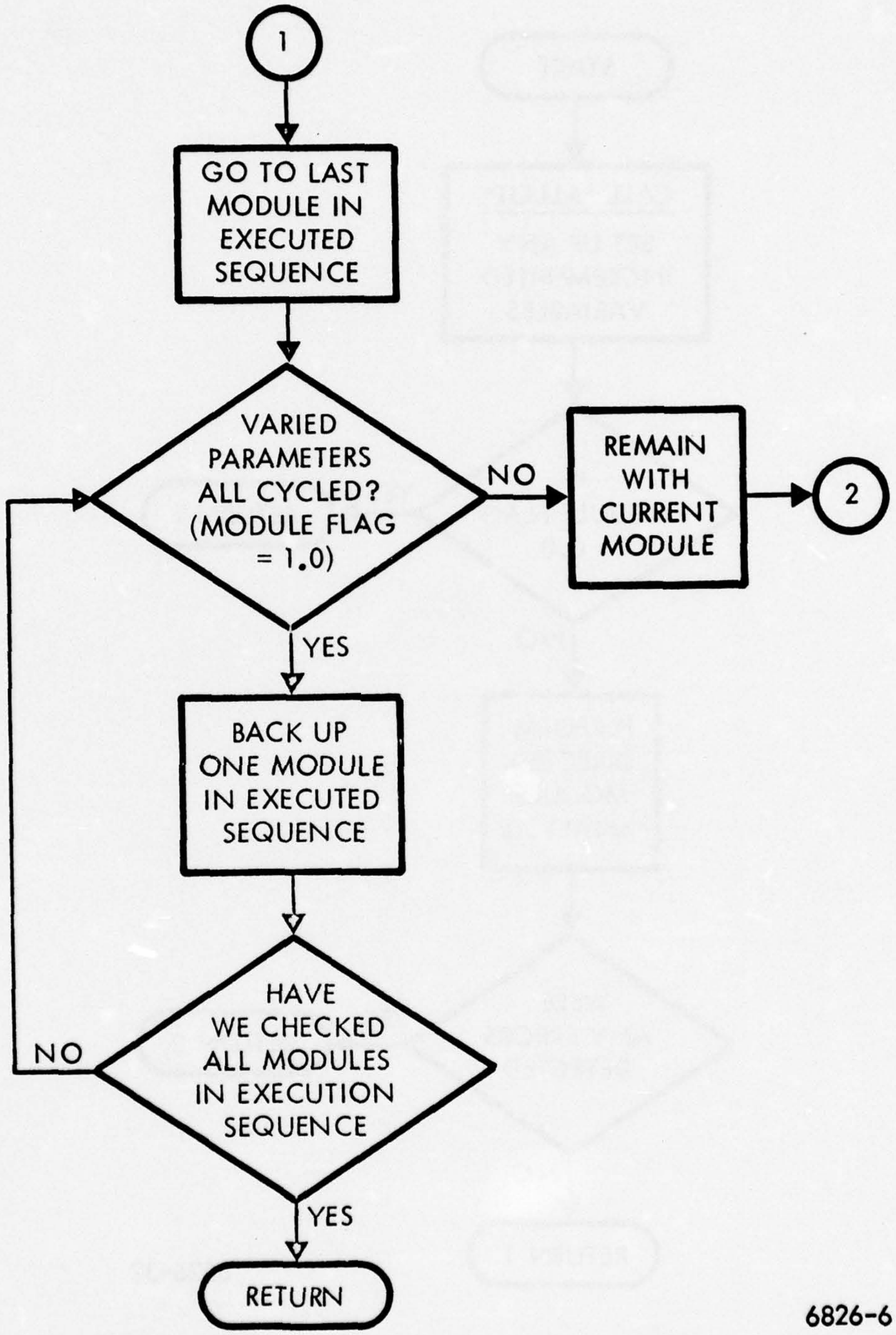
Logic Diagram for TAMDAT, FAMDAT, RAMDAT, and SAMDAT (Sheet 2 of 3)



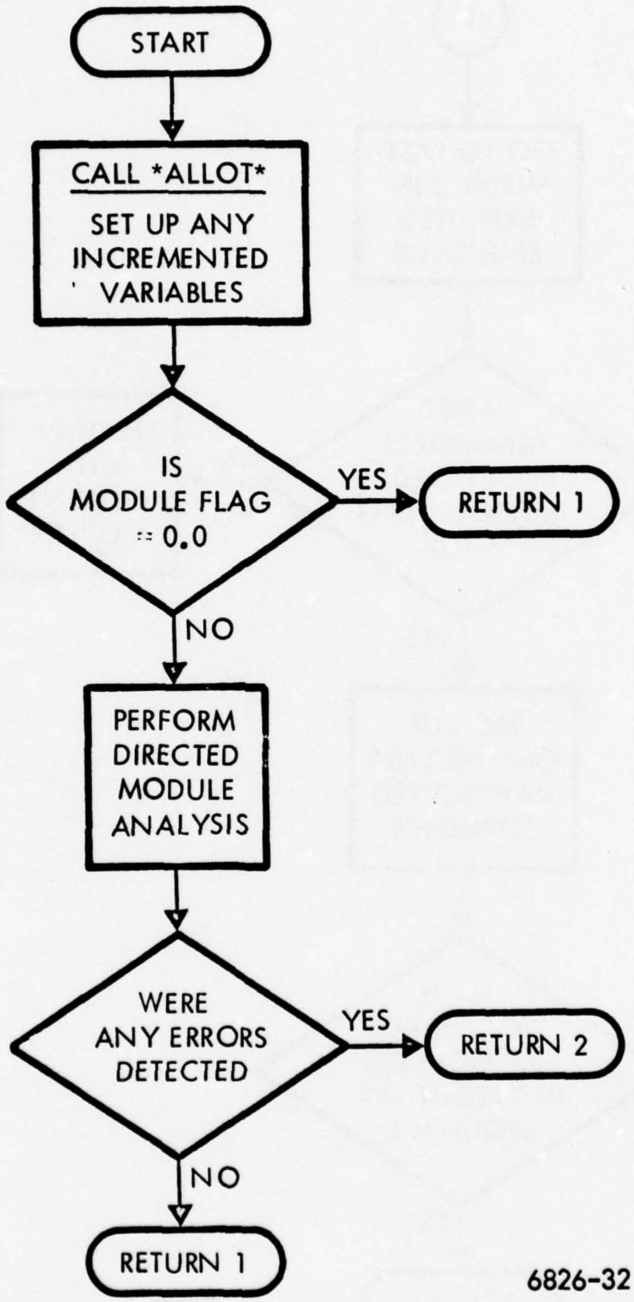
Logic Diagram for TAMDAT, FAMDAT, RAMDAT, and SAMDAT (Sheet 3 of 3)



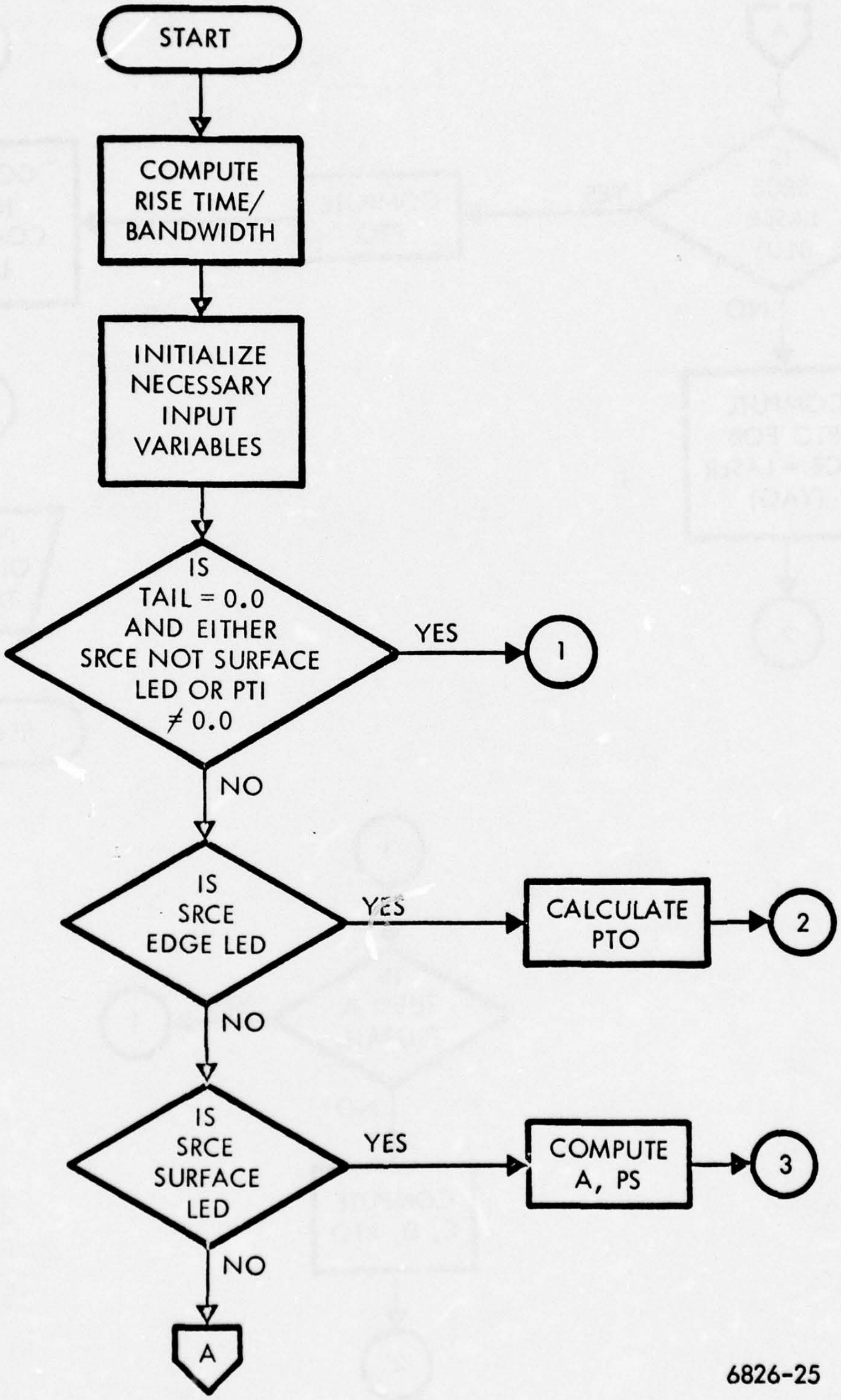
Logic Diagram for SYS (Sheet 1 of 2)



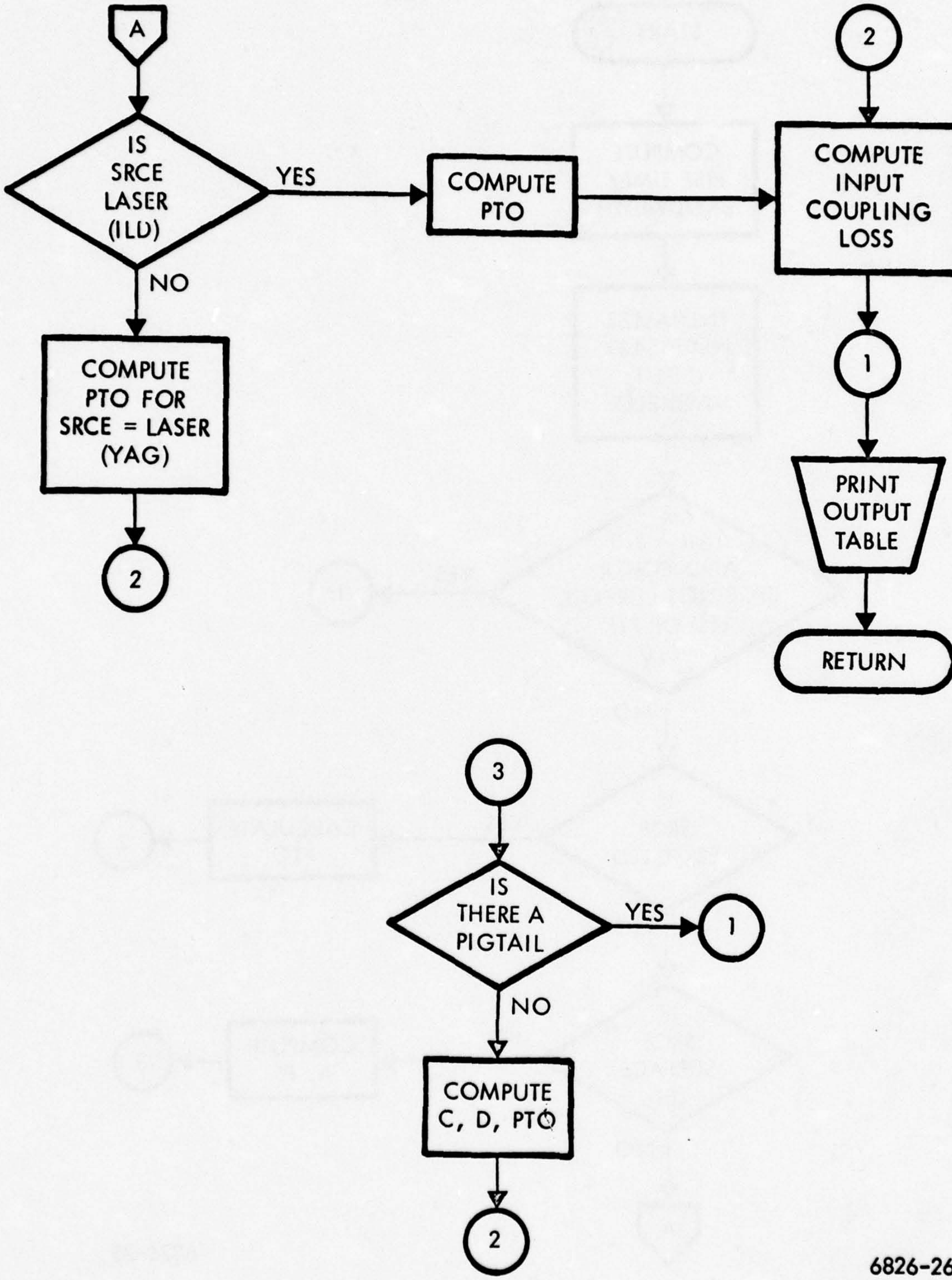
Logic Diagram for SYS (Sheet 2 of 2)



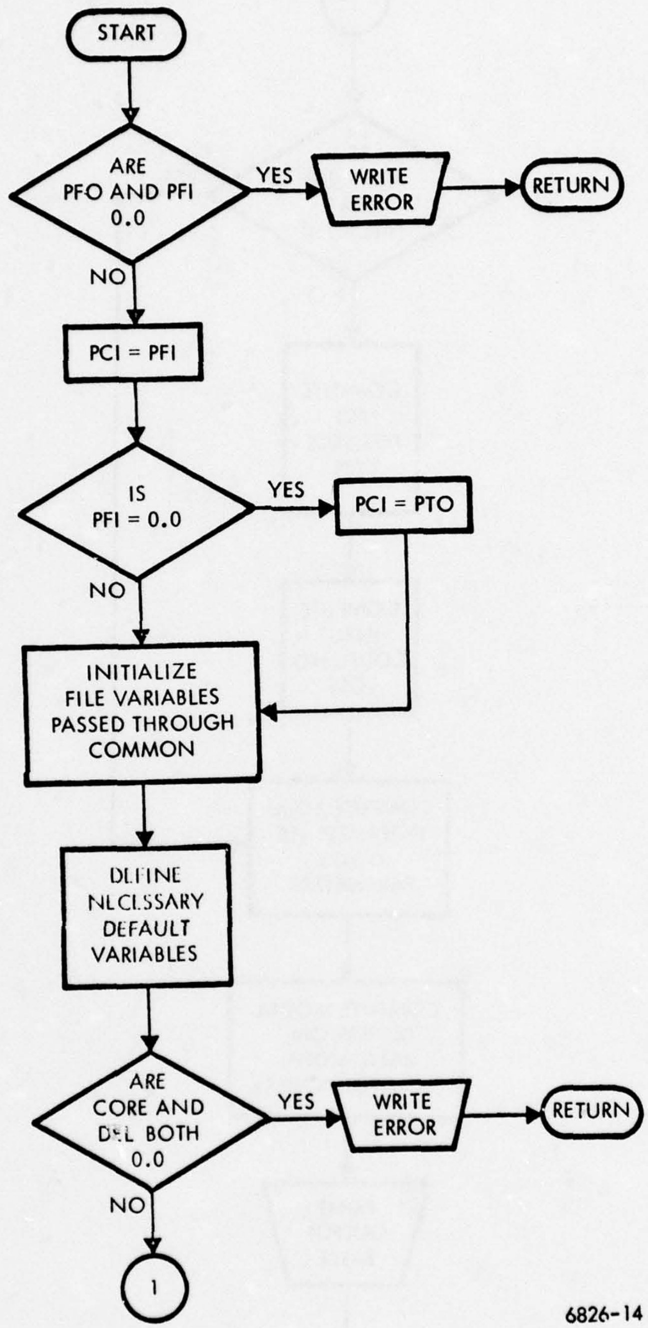
Logic Diagram for ETAM, EFAM, ERAM, and ESAM



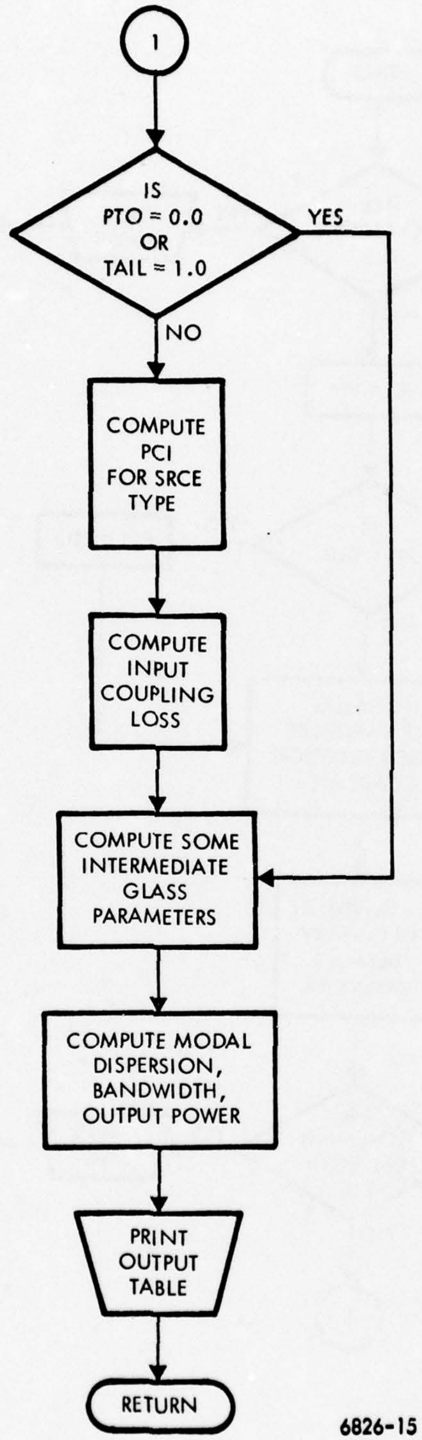
Logic Diagram for TAM (Sheet 1 of 2)



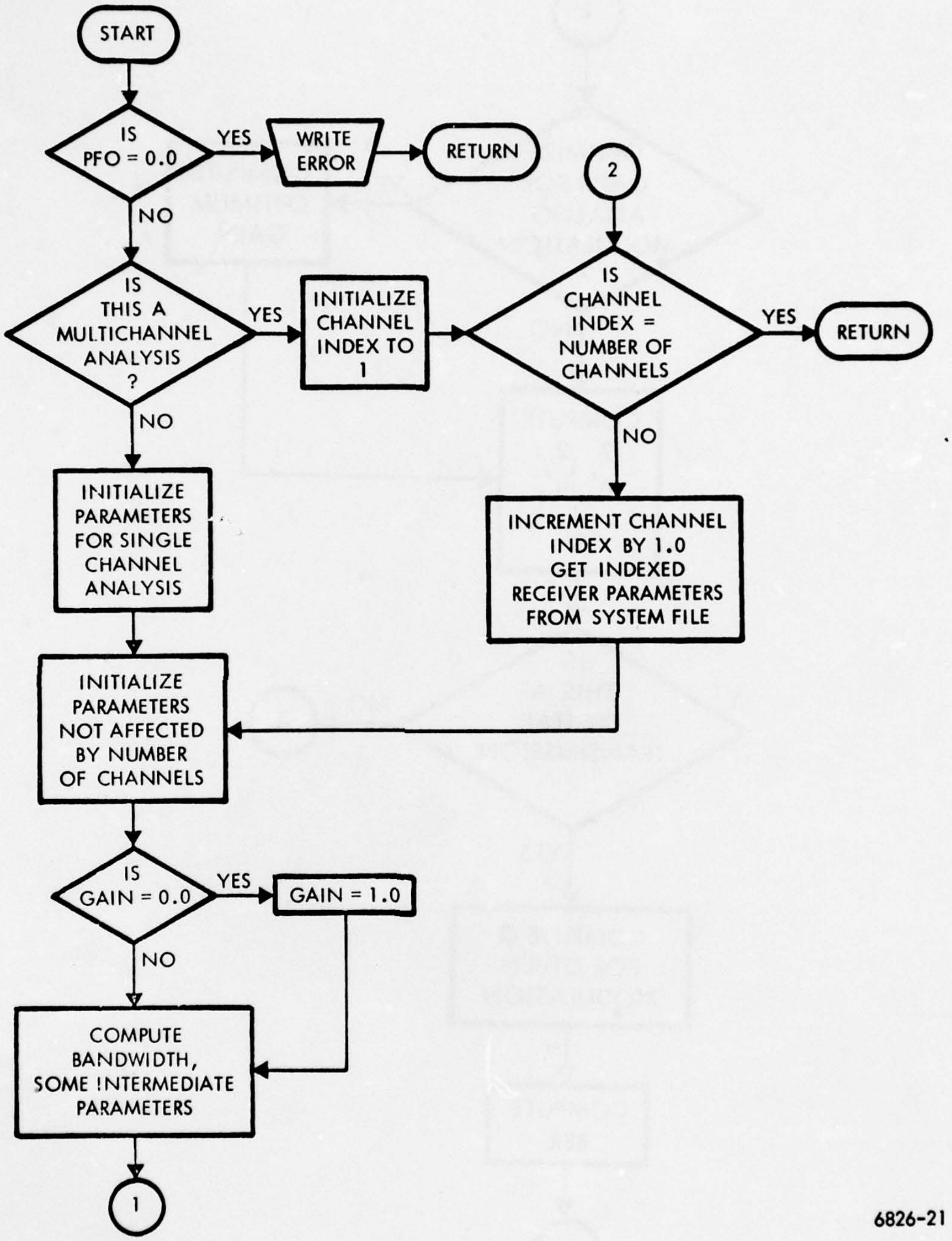
Logic Diagram for TAM (Sheet 2 of 2)



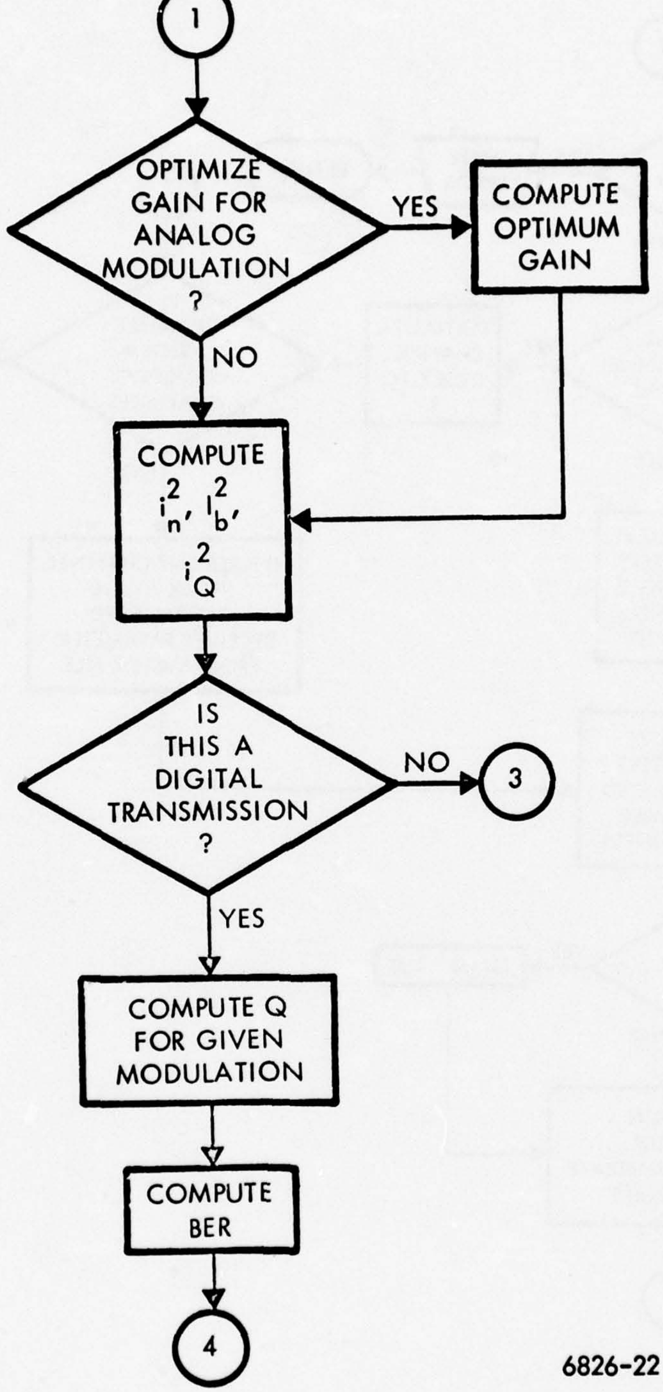
Logic Diagram for FAM (Sheet 1 of 2)



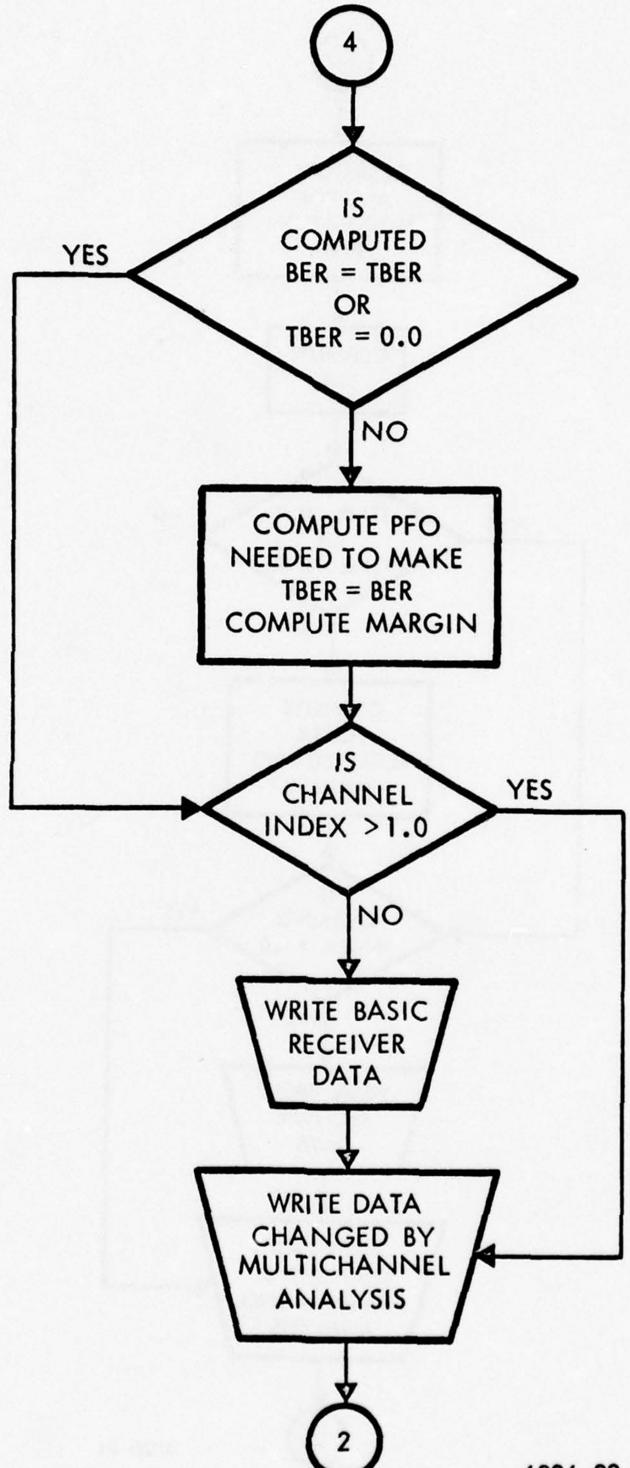
Logic Diagram for FAM (Sheet 2 of 2)



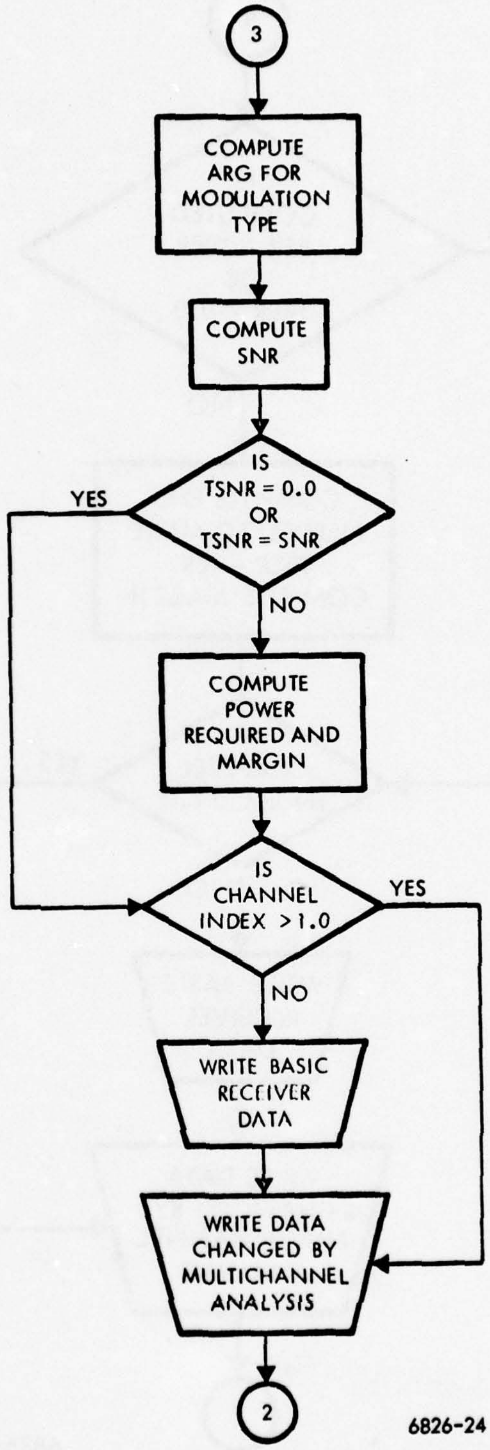
Logic Diagram for RAM (Sheet 1 of 4)



Logic Diagram for RAM (Sheet 2 of 4)



Logic Diagram for RAM (Sheet 3 of 4)



Logic Diagram for RAM (Sheet 4 of 4)

3.13 Program Listings

The following pages contain a complete set of listings of the FODAP routines. The routines are grouped by functions; the EXEC, the MANAGER, TAM, FAM, RAM, and SAM.

```
      DOUBLE PRECISION NA(1),NTSK(20),BLANK
      COMMON /ERR/NFLAG(10)
      COMMON /LABEL/LAB(25,3)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      DATA BLANK/8H        /
      EATA LAB(1,1)/4HCORE/
      DATA NTSK/8HCREATE  ,8HGET     ,8HSAVE    ,8HLIST    ,
\C8HDELETE  ,
      8HDEFINE  ,8HPWRITE  ,8HFWRITE  ,8HFAM     ,8HRAM
\C,8HTAM     ,
      8HSAM     ,8HINPUT   ,8HEND     ,8HPLOT    ,8HORDER
\C,8HSYS     ,
      8HCLEAR   ,8H        ,8H        /
      DATA IND/1HN/
C     SET UNIT NOS,ZERO FLAGS
      CALL BLOCK
      NCMD=18
      B=LAB(1,1)
      NCRD=7
      NTER=1
      LDI=5
      LDO=6
      LDF=17
      LDA=4
      CALL CLEAR
      WRITE(LDO,2000)
 2000 FORMAT(///,"0**FODAP EXECUTIVE**",/,"    READY FOR INPU
\CT")
   21 IF(NFLAG(2).EQ.1)STOP ERROR
      CALL CONCRD
      CALL DCODE(IND,NBEG,NEND,NLEN,$9000)
      IF(NLEN.GT.9)GO TO 9000
      NA(1)=BLANK
      ENCODE(NA(1),1000)(LINE(K),K=NBEG,NEND)
      DO 100 I=1,NCMD
  251 FORMAT(1X,A8,2X,A8)
      IF(NA(1).EQ.NTSK(I))GO TO 300
  100 CONTINUE
      GO TO 9010
  300 GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
\C,20),I
    1 CALL CREATE
      GO TO 21
    2 CALL GET
      GO TO 21
    3 CALL SAVE
      GO TO 21
    4 CALL LIST
      GO TO 21
    5 CALL DELETE
      GO TO 21
    6 CALL DEFINE
      GO TO 21
```

```
    7 CALL PWRITE
      GO TO 21
    8 CALL FWRITE
      GO TO 21
    9 CALL SET(1)
      GO TO 21
   10 CALL SET(2)
      GO TO 21
   11 CALL SET(3)
      GO TO 21
   12 CALL SET(4)
      GO TO 21
   13 CALL INPUTD
      GO TO 21
   14 STOP
   15 CALL CPLOT
      GO TO 21
   16 CALL ORDER
      GO TO 21
   17 CALL SYS
      GO TO 21
   18 CALL CLEAR
      GO TO 21
   19 CONTINUE
   20 CONTINUE
      GO TO 21
 1000 FORMAT(72A1)
 9000 CALL ERROR(1)
      GO TO 21
 9010 CALL ERROR(2)
      GO TO 21
      END
```

```
      SUBROUTINE CONCRD
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
    1 READ(LDI,1000,END=10000)LINE
 1000 FORMAT(72A1)
      RETURN
10000 GO TO 1
      END
```

```
      SUBROUTINE DCODE(IND,NBEG,NEND,NLEN,*)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      INTEGER BLANK,COMMA,FLAG,EQUAL
      DATA BLANK,COMMA,NN/1H ,1H,,1HN/,EQUAL/1H=/
      DATA OPAR,CPAR/1H(,1H)/
      NBEG=1
      NEND=72
      IF(IND.NE.NN)GO TO 2
      N=NBEG
      M=NEND
    2 FLAG=0
      DO 100 I=N,M
      IF(FLAG.EQ.0.AND.LINE(I).EQ.BLANK)GO TO 100
      IF(FLAG.EQ.1.AND.(LINE(I).EQ.BLANK.OR.LINE(I).EQ.COMMA
\C.OR.
      LINE(I).EQ.EQUAL.OR.LINE(I).EQ.OPAR.OR.LINE(I).EQ.CPAR
\C))GO TO 3
      IF(FLAG.EQ.1)GO TO 100
      FLAG=1
      NBEG=I
  100 CONTINUE
    4 RETURN 1
    3 IF(I.NE.M)GO TO 1
      IF(LINE(I).NE.EQUAL)GO TO 4
      N=I+1
      GO TO 2
    1 NEND=I-1
      NLEN=NEND-NBEG+1
      N=I+1
      M=72
      RETURN
      END
```

```
      SUBROUTINE SECFIL(NA,I,*)
      DIMENSION NA(2),NSEC(5,2),LA(2)
      DIMENSION MSEC(4),LSEC(4)
      DATA LSEC/4HFAM ,4HRAM ,4HTAM ,4HSAM /
      DATA NSEC/4HFAMF,4HRAMF,4HTAMF,4HSAMF,4HNEWF,4HILE ,4H
\FILE ,
      4HILE ,4HILE ,4HILE /
      DATA MSEC/4HFAMD,4HRAMD,4HTAMD,4HSAMD/
      DO 100 I=1,5
      IF(NA(1).NE.NSEC(I,1))GO TO 100
      IF(NA(2).EQ.NSEC(I,2))RETURN
  100 CONTINUE
      RETURN 1
      ENTRY MODULE(MA,I,*)
      DO 200 I=1,4
      IF(MA.EQ.MSEC(I))RETURN
  200 CONTINUE
      RETURN 1
      ENTRY MODA(LA,I,*)
      DO 300 I=1,4
      IF(LA.EQ.LSEC(I))RETURN
  300 CONTINUE
      RETURN 1
      END
```

```
      SUBROUTINE SYS
      DIMENSION NFL(4)
      COMMON /SYS/ORDR(4),JEND
      COMMON /ERR/NFLAG(10)
      NFLAG(3)=0
      NFLAG(4)=0
      NFLAG(5)=0
      I=1
    5 IF(ORDR(I).EQ.1.0)CALL EFAM($4,NFL(I))
      IF(ORDR(I).EQ.2.0)CALL ERAM($4,NFL(I))
      IF(ORDR(I).EQ.3.0)CALL ETAM($4,NFL(I))
      IF(ORDR(I).EQ.4.0)CALL ESAM($4)
      IF(I.EQ.JEND)GO TO 1
      IF(NFL(I).EQ.0)GO TO 1
      I=I+1
      GO TO 5
    1 DO 10 K=JEND,1,-1
      IF(NFL(K).EQ.1)GO TO 5
      I=K-1
   10 CONTINUE
    4 RETURN
      END
```

```
      SUBROUTINE ALLOT(N,X,L,*)
      COMMON /ERR/NFLAG(10)
      COMMON /INCT/XINC(10,7),CT(5),CNT
      DIMENSION X(L)
      IF(CT(N).EQ.0.0)GO TO 9997
      IF(NFLAG(N+2).EQ.0)GO TO 9998
      XN=N
      SKIP=10000.0
    2 XNEXT=0.0
      IND=0
    1 DO 10 I=1,10
      IF(XINC(I,2).NE.XN.OR.SKIP.LE.XINC(I,1).OR.XNEXT.GT.XI
     \NC(I,1))
      GO TO 10
      XNEXT=XINC(I,1)
      IND=I
   10 CONTINUE
      I=IND
      IF(IND.EQ.0)GO TO 9996
      IF(XINC(I,6).GT.0.0.AND.(XINC(I,7).EQ.XINC(I,5).OR.(XI
     \NC(I,7)+
      XINC(I,6)).GT.XINC(I,5)))GO TO 100
      IF(XINC(I,6).LT.0.0.AND.(XINC(I,7).EQ.XINC(I,5).OR.(XI
     \NC(I,7)+
      XINC(I,6)).LT.XINC(I,5)))GO TO 100
      XINC(I,7)=XINC(I,7)+XINC(I,6)
      K=XINC(I,3)
      X(K)=XINC(I,7)
      GO TO 9999
  100 XINC(I,7)=XINC(I,4)
      SKIP=XINC(I,1)
      K=XINC(I,3)
      X(K)=XINC(I,7)
      GO TO 2
C
C     WE ARE FINISHED
 9997 NFLAG(N+2)=2
      GO TO 9999
 9998 NFLAG(N+2)=1
 9999 RETURN
 9996 NFLAG(N+2)=0
      RETURN 1
      END
```

```
SUBROUTINE SET(N)
COMMON /SYS/ORDR(4),JEND
SAVE=ORDR(1)
K=JEND
JEND=1
ORDR(1)=N
CALL SYS
ORDR(1)=SAVE
JEND=K
RETURN
END
```

```
      SUBROUTINE DATCAL(I,IOPT,*)
      IF(I.EQ.1)CALL FAMDAT(IOPT,$9999)
      IF(I.EQ.2)CALL RAMDAT(IOPT,$9999)
      IF(I.EQ.3)CALL TAMDAT(IOPT,$9999)
      IF(I.EQ.4)CALL SAMDAT(IOPT,$9999)
      RETURN
 9999 RETURN 1
      END
```

```
      SUBROUTINE INPUTD
      DOUBLE PRECISION NAME(1),NCARD,IACT,NCARDS,NBLNK
      COMMON /ERR/NFLAG(10)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      DATA NCARD,NCARDS,IACT/8HCARD    ,8HCARDS   ,8HIACT
 \C /
      DATA D/1HD/,NBLNK/8H        /
      CALL DCODE(D,NBEG,NEND,NLEN,$9000)
      IF(NLEN.GT.5)GO TO 9000
      NAME(1)=NBLNK
      ENCODE(NAME,1000)(LINE(K),K=NBEG,NEND)
      IF(NAME(1).NE.NCARD.AND.NAME(1).NE.NCARDS)GO TO 1
      LDI=7
      NFLAG(1)=1
      GO TO 9999
    1 IF(NAME(1).NE.IACT)GO TO 9000
      LDI=5
      NFLAG(1)=0
 9999 RETURN
 1000 FORMAT(72A1)
 9000 CALL ERROR(1)
      RETURN
      END
```

```
      SUBROUTINE CLEAR
      DIMENSION TRAN(30,5)
      COMMON /ERR/NFLAG(10)
      COMMON /SYS/ORDR(4),JEND
      COMMON /INCT/XINC(10,7),CT(5),CNT
      COMMON /FAMS/A(31)
      COMMON /TAMS/C(14)
      COMMON /RAMS/B(23)
      COMMON /SPECL/TRAN,DPR
      K=NFLAG(1)
      DO 10 I=1,10
      XINC(I,2)=0.0
   10 NFLAG(I)=0
      NFLAG(1)=K
      ORDR(1)=3.0
      ORDR(2)=1.0
      ORDR(3)=2.0
      ORDR(4)=4.0
      JEND=4
      CNT=0.0
      DPR=0.0
      DO 20 I=1,5
      DO 25 J=1,30
   25 TRAN(J,I)=0.0
   20 CT(I)=0.0
      DO 30 I=1,31
   30 A(I)=0.0
      DO 40 I=1,14
   40 C(I)=0.0
      DO 50 I=1,22
   50 B(I)=0.0
      RETURN
      END
```

```
      SUBROUTINE ORDER
      INTEGER BLANK
      DIMENSION NA(1)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /ERR/NFLAG(10)
      COMMON /SYS/ORDR(4),JEND
      DATA BLANK/4H    /,D/4HD   /
      J=0
C    GET FIRST MODULE
   10 NA(1)=BLANK
      CALL DCODE(D,NBEG,NEND,NLEN,$9999)
      IF(NLEN.GT.3.OR.(J+1).GT.4)GO TO 9010
      J=J+1
      ENCODE(NA(1),1000) (LINE(K),K=NBEG,NEND)
C     IDENTIFY MODULE
      CALL MODA(NA(1),I,$9010)
      ORDR(J)=I
      JEND=J
      GO TO 10
 1000 FORMAT(72A1)
 9010 CALL ERROR(25)
 9999 RETURN
      END
```

```
      SUBROUTINE ERROR(N)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /ERR/NFLAG(10)
      IF(NFLAG(1).EQ.1)NFLAG(2)=1
      IF(N.GT.20)GO TO 100
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
     C,20),N
  100 M=N-20
      GO TO (21,22,23,24,25,26,27,28,29,30),M
    1 WRITE(LDO,9001)
      GO TO 10000
    2 WRITE(LDO,9002)
      GO TO 10000
    3 WRITE(LDO,9003)
      GO TO 10000
    4 WRITE(LDO,9004)
      GO TO 10000
    5 WRITE(LDO,9005)
      GO TO 10000
    6 WRITE(LDO,9006)
      GO TO 10000
    7 WRITE(LDO,9007)
      GO TO 10000
    8 WRITE(LDO,9008)
      GO TO 10000
    9 WRITE(LDO,9009)
      GO TO 10000
   10 WRITE(LDO,9010)
      GO TO 10000
   11 WRITE(LDO,9011)
      GO TO 10000
   12 WRITE(LDO,9012)
      GO TO 10000
   13 WRITE(LDO,9013)
      GO TO 10000
   14 WRITE(LDO,9014)
      GO TO 10000
   15 WRITE(LDO,9015)
      GO TO 10000
   16 WRITE(LDO,9016)
      GO TO 10000
   17 WRITE(LDO,9017)
      GO TO 10000
   18 WRITE(LDO,9018)
      GO TO 10000
   19 WRITE(LDO,9019)
      GO TO 10000
   20 WRITE(LDO,9020)
      GO TO 10000
   21 WRITE(LDO,9021)
      GO TO 10000
   22 WRITE(LDO,9022)
      GO TO 10000
```

```
   23 WRITE(LDO,9023)
      GO TO 10000
   24 WRITE(LDO,9024)
      GO TO 10000
   25 WRITE(LDO,9025)
      GO TO 10000
   26 WRITE(LDO,9026)
      GO TO 10000
   27 WRITE(LDO,9027)
      GO TO 10000
   28 WRITE(LDO,9028)
      GO TO 10000
   29 WRITE(LDO,9029)
      GO TO 10000
   30 WRITE(LDO,9030)
10000 RETURN
 9001 FORMAT(" ERROR**IMPROPER FORMAT FOR COMMAND LINE**")
 9002 FORMAT(" ERROR**COMMAND DOES NOT EXIST**")
 9003 FORMAT(" ERROR**FILENAME MORE THAN 4 CHARACTERS**")
 9004 FORMAT(" ERROR**FILE-TYPE INCORRECTLY ENTERED**")
 9005 FORMAT(" ERROR**DEFINE STATEMENT INCOMPLETE**")
 9006 FORMAT(" ERROR**ERROR ON TAMFILE CARD READ**")
 9007 FORMAT(" ERROR**DUPLICATE NAME IN DIRECTORY**")
 9008 FORMAT(" ERROR**ERROR ON FAMFILE CARD READ**")
 9009 FORMAT(" ERROR**DIRECTORY NAME IS INCORRECT**")
 9010 FORMAT(" ERROR**FILE-NAME HAS NOT BEEN GIVEN**")
 9011 FORMAT(" ERROR**DIRECTORY IS FULL**")
 9012 FORMAT(" ERROR**NEWFILE IS NOT A VALID FILE-TYPE**")
 9013 FORMAT(" ERROR**INCREMENT TABLE IS FULL**")
 9014 FORMAT(" ERROR**THIS DIRECTORY NAME IS ILLEGAL**")
 9015 FORMAT(" ERROR**FILE-NAME DOES NOT EXIST**")
 9016 FORMAT(" ERROR**TOO MANY CHARACTERS**")
 9017 FORMAT(" ERROR**FILES HAVE NOT BEEN INITIALIZED**")
 9018 FORMAT(" ERROR**ZERO INCREMENT VALUE ILLEGAL**")
 9019 FORMAT(" ERROR**UNIDENTIFIED VARIABLE NAME**")
 9020 FORMAT(" ERROR**VALUE NOT ENTERED CORRECTLY**")
 9021 FORMAT(" ERROR**ERROR ON RAMFILE CARD READ**")
 9022 FORMAT(" ERROR**NOT ENOUGH INPUT DATA FOR TAM MODULE**
     C")
 9023 FORMAT(" ERROR**INCORRECT SRCE TYPE SPECIFIED**")
 9024 FORMAT(" ERROR**VARIABLE VALUE OUTSIDE ALLOWABLE RANGE
     C**")
 9025 FORMAT(" ERROR**INCORRECT NAME**")
 9026 FORMAT(" ERROR**")
 9027 FORMAT(" ERROR**")
 9028 FORMAT(" ERROR**")
 9029 FORMAT(" ERROR**")
 9030 FORMAT(" ERROR**")
      END
```

```
      SUBROUTINE DIRECT(NTYPE,NAMEP,IACT,*,*)
      DIMENSION ISNO(99),NA(2),INAME(99)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /ERR/NFLAG(10)
      COMMON NF,IFADD
      DATA IBL,INOF,NYES,NO/4H     ,99,4HYES ,4HNO  /
C     NTYPE - INDICATES FILE-TYPE
C          1 - FIBER FILE
C          2- RECEIVER FILE
C          3 - TRANSMITTER FILE
C          4 - SYSTEM FILE
C          6 - ALL FILES
C
C     NAME   - FILE OF DIRECTORY NAME
C
C      IACT - ACTION INDICATOR
C
C          1 - CREATE NEW DIRECTORY
C          2 - ENTER NAME IN DIRECTORY
C          3 - CHECK NAME FOR DUPLICATION
C          4 - LOOK UP  NAME IN DIRECTORY
C          5 - DELETE NAME FROM DIRECTORY
C          6 - GET ADDRESS FOR NEW FILE
C          7 - LIST DIRECTORY
C
      NAME=NAMEP
      IF(IACT.EQ.2)GO TO 3
      IF(IACT.EQ.5)GO TO 13
      IF(IACT.NE.1)GO TO 6
C     INITIALIZE DIRECTORIES
      DO 100 I=1,INOF
      INAME(I)=IBL
  100 ISNO(I)=9999
      IEND=0
      IF(NFLAG(1).EQ.1)GO TO 2
      WRITE(LDO,8000)
 8000 FORMAT(1X,"****YOU ARE ABOUT TO DESTROY ANY EXISTING F
\CILES.",/,
      1X,"****ENTER YES TO CONTINUE, NO TO ABORT COMMAND.")
    1 READ(LDI,1000)NANS
 1000 FORMAT(A4)
      IF(NANS.EQ.NO)RETURN 2
      IF(NANS.EQ.NYES)GO TO 2
      WRITE(LDO,8001)
 8001 FORMAT(1X,"PLEASE ANSWER YES OR NO. NO WILL RETURN YOU
\C TO CONTROL
      MODE.")
      GO TO 1
    2 IF(NTYPE.EQ.6.OR.NTYPE.EQ.4)WRITE(LDF'301)IEND,(INAME(
\CK)
      ,ISNO(K),K=1,INOF)
      IF(NTYPE.EQ.6.OR.NTYPE.EQ.3)WRITE(LDF'201)IEND,(INAME(
```

```
\(K)
      ,ISNO(K),K=1,INOF)
      IF(NTYPE.EQ.6.OR.NTYPE.EQ.2)WRITE(LDF'1)IEND,(INAME(K)
     .ISNO(K),K=1,INOF)
      IF(NTYPE.EQ.6.OR.NTYPE.EQ.1)WRITE(LDF'101)IEND,(INAME(
\(K)
     .ISNO(K),K=1,INOF)
      GO TO 9999
    6 IF(NTYPE.EQ.4)NSEC=301
      IF(NTYPE.EQ.3)NSEC=201
      IF(NTYPE.EQ.2)NSEC=1
      IF(NTYPE.EQ.1)NSEC=101
      READ(LDF'NSEC)IEND,(INAME(I),ISNO(I),I=1,INOF)
      GO TO (9999,3,4,4,4,8,14),IACT
    3 INAME(NREC)=NAME
      ISNO(NREC)=IFADD
      GO TO 10
    7 INAME(I)=NAME
      ISNO(I)=NSEC+I+1
      NREC=ISNO(I)
   12 WRITE(LDF'NSEC)IEND,(INAME(I),ISNO(I),I=1,INOF)
C     CHECK NAME FOR DUPLICATION
    4 DO 300 I=1,IEND
      IF(NAME.EQ.INAME(I))GO TO 5
  300 CONTINUE
      IF(IACT.EQ.4)GO TO 9998
      IF(IACT.EQ.5)GO TO 9050
C     NAME DOES NOT EXIST
      GO TO 9999
C     NAME HAS BEEN FOUND
    5 IF(IACT.EQ.3)RETURN 1
      IF(IACT.EQ.5)GO TO 13
      NREC=I
       IFADD=ISNO(I)
      GO TO 9999
C     READY TO DELETE FILE
   13 ISNO(NREC)=9999
      INAME(NREC)=IBL
      GO TO 10
   14 DO 200 I=1,IEND
      IF(ISNO(I).NE.9999)WRITE(LDO,1001)INAME(I)
  200 CONTINUE
      GO TO 9999
C     FIND AN EMPTY CELL TO ENTER NAME
    8 DO 400 I=1,INOF
      IF(ISNO(I).EQ.9999)GO TO 9
  400 CONTINUE
      GO TO 9998
    9 NREC=I
      IFADD=NSEC+NREC+1
      GO TO 9999
C     SET LAST CELL FOR IEND
```

```
   10 DO 500 I=INOF,1,-1
      IF(ISNO(I).NE.9999)GO TO 11
  500 CONTINUE
      IEND=INOF
      WRITE(LDO,8002)
       GO TO 120
   11 IEND=I
C     WRITE DIRECTORY
  120 WRITE(LDF'NSEC)IEND,(INAME(K),ISNO(K),K=1,INOF)
      GO TO 9999
 1001 FORMAT(1X,A4)
 8002 FORMAT(1X,"****WARNING**** THIS DIRECTORY IS NOW FULL
 *(*")
 9050 CALL ERROR(15)
 9998 RETURN 1
 9999 RETURN
      END
```

```
      SUBROUTINE EFAM(*,N)
      COMMON /ERR/NFLAG(10)
      COMMON /FAMS/A(31)
      COMMON /TAMS/C(14)
      COMMON /RAMS/B(23)
      COMMON /INCT/XINC(10,7),CT(5),CNT
      CALL ALLOT(1,A,31,$1)
      CALL FAM($9998)
    1 N=NFLAG(3)
      GO TO 9999
      ENTRY ETAM(*,N)
      CALL ALLOT(3,C,14,$2)
      CALL TAM($9998)
    2 N=NFLAG(5)
      GO TO 9999
      ENTRY ERAM(*,N)
      CALL ALLOT(2,B,23,$3)
      CALL RAM($9998)
    3 N=NFLAG(4)
      GO TO 9999
      ENTRY ESAM(*)
      CALL SAM($9998)
 9999 RETURN
 9998 RETURN 1
      END
```

```
      SUBROUTINE TAMDAT(NOPT,*)
C     THIS SUBROUTINE CALLED BY:  CREATE,LIST,GET,SAVE
C
C     OPTIONS ARE:
C          1 - READ INITIAL DATA AND WRITE TO FILE
C          2 - READ DATA FROM FILE
C          3 - WRITE A FILE LISTING
C          4 - LIST CURRENT DATA
C          5 - WRITE CURRENT DATA TO FILE
C
      COMMON /ERR/NFLAG(10)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /TAMS/TAIL,XLEN,XIO,BTR,PTI,SRCE,RT,ANGL,DS,BW,
&CTNA,
      TFP,TDF,RTI
      COMMON /FAMS/CORE,DF,FP,DEL,XLAM,SPEC,ALFA,XLMC,CLAD,X
&CNU,PN1
      ,PDD,PSEC,AD,AF,AS,CN,SN,XL,XAI,XAO,PFI,PTO,
      CONN,SPLC,FIBR,TRA,TER,TOT,BL,GAI
      COMMON NAME,IFADD
      GO TO (1,2,3,4,5),NOPT
C     READ INITIAL DATA FROM CARDS OR TELETYPE
C
C     INITIALIZE DATA
C
    1 DS=0.0
      ANGL=0.0
      XIO=0.0
      BTR=0.0
      TNA=0.0
      TFP=0.0
      TDF=0.0
      IF(NFLAG(1).NE.1)GO TO 10
      READ(LDI,ERR=9000)SRCE,TAIL,XLEN,PTI,ANGL,DS,XIO,BTR,R
&CT,BW,
      SPEC,XLAM,TNA,TFP,TDF
      GO TO 5
   10 WRITE(LDO,8000)
      NWALK=1
  100 READ(LDI,ERR=9001,END=9999)SRCE,TAIL,XLEN,PTI
      WRITE(LDO,8030)
      NWALK=4
  130 READ(LDI,ERR=9001,END=9999)XLAM,SPEC,RT,BW
      IF(TAIL.EQ.0.0)GO TO 6
      WRITE(LDO,8040)
      NWALK=5
  140 READ(LDI,ERR=9001,END=9999)TNA,TFP,TDF
    6 IF(SRCE.EQ.1.0.OR.SRCE.EQ.3.0)GO TO 5
      IF(SRCE.EQ.4.0)GO TO 7
      IF(SRCE.NE.2.0)GO TO 9004
      WRITE(LDO,8010)
      NWALK=2
```

```
  110 READ(LDI,ERR=9001,END=9998)ANGL,DS,XID,BTR
      GO TO 5
    7 WRITE(LDO,8020)
      NWALK=3
  120 READ(LDI,ERR=9001,END=9998)DS
    5 WRITE(LDF'IFADD)SRCE,TAIL,XLEN,PTI,ANGL,DS,XID,BTR,RT,
\CBW,SPEC,XLAM
      ,TNA,TFP,TDF
      GO TO 9999
    3 CALL DIRECT(3,NAME,4,$9998,$9999)
    2 READ(LDF'IFADD)SRCE,TAIL,XLEN,PTI,ANGL,DS,XID,BTR,RT,B
\CW,SPEC,XLAM
      ,TNA,TFP,TDF
      IF(NOPT.EQ.2)GO TO 9999
    4 WRITE(LDO,8001,ERR=9003)SRCE,TAIL,XLEN,PTI,ANGL,DS,XID
\C,BTR,RT,BW,
      SPEC,XLAM,TNA,TFP,TDF
      GO TO 9999
 8000 FORMAT(" ENTER VALUES FOR: SRCE,TAIL,XLEN,PTI")
 8010 FORMAT(" ENTER VALUES FOR: ANGL,DS,XID,BTR")
 8020 FORMAT(" ENTER A VALUE FOR: DS")
 8030 FORMAT(" ENTER VALUES FOR: XLAM,SPEC,RT,BW")
 8040 FORMAT(" ENTER VALUES FOR: TNA,TFP,TDF")
 8001 FORMAT(" SRCE=",E11.5,T24,"TAIL=",E11.5,T48,"XLEN=",E1
\C1.5,/,
      "  PTI=",E11.5,T24,"ANGL=",E11.5,T48,"  DS=",E11.5,/,"
\C  XID=",
      E11.5,T24," BTR=",E11.5,T48,"  RT=",E11.5,/,"   BW=",E
\C11.5,T24
      ,"SPEC=",E11.5,T48,"XLAM=",E11.5,/,"  TNA=",E11.5,T24,
\C" TFP=",
      E11.5,T48," TDF=",E11.5,/," *END FILE*")
 9000 CALL ERROR(6)
      GO TO 9998
 9001 WRITE(LDO,8002)
 8002 FORMAT("0**INPUT ERROR.  PLEASE TRY AGAIN**")
      GO TO (100,110,120,130,140),NWALK
 9003 CALL ERROR(17)
      GO TO 9998
 9004 CALL ERROR(23)
 9998 RETURN 1
 9999 RETURN
      END
```

```
      SUBROUTINE FAMDAT(NOPT,*)
C     THIS SUBROUTINE CALLED BY:  CREATE,LIST,GET,SAVE
C
C     OPTIONS ARE:
C         1 - READ INITIAL DATA AND WRITE TO FILE
C         2 - READ DATA FROM FILE
C         3 - WRITE A FILE LISTING
C         4 - LIST CURRENT DATA
C         5 - WRITE CURRENT DATA TO FILE
C
      COMMON /ERR/NFLAG(10)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /TAMS/TAIL,XLEN,XID,BTR,PTI,SRCE,RT,ANGL,DS,BW,
\CTNA,TFP,TDF
      COMMON /FAMS/CORE,DF,FP,DEL,XLAM,SPEC,ALFA,XLMC,CLAD,X
\CNA,PN1
      ,PDD,PSEC,AC,AF,AS,CN,SN,XL,XAI,XAO,PFI,PTO,
      CONN,SPLC,FIBR,TRA,TER,TOT,BL,GAI
      COMMON /RAMS/XAD,XIL,XIB,XISQ,RES,XM,SPMS,RNB,XMSC,XMA
\C,BETA,R,
      TYPE,OPT,GAIN,TBER,TSNR,PFO,CHAN,
      GOPT,BER,SNR,BAND
      COMMON NAME,IFADD
      GO TO (1,2,3,4,5),NOPT
C     READ INITIAL DATA FROM CARDS OR TELETYPE
    1 IF(NFLAG(1).NE.1)GO TO 10
      READ(LDI,ERR=9000)CORE,DF,FP,DEL,ALFA,XLMC,CLAD,XNA,
      PN1,PDD,PSEC,AF,AS,AC,XAI
      GO TO 5
   10 WRITE(LDO,8000)
      NWALK=1
  100 READ(LDI,ERR=9001,END=9998)CORE,DF,FF,DEL
      WRITE(LDO,8020)
      NWALK=3
  120 READ(LDI,ERR=9001,END=9998)ALFA,XLMC,CLAD,XNA
      WRITE(LDO,8030)
      NWALK=4
  130 READ(LDI,ERR=9001,END=9998)PN1,PDD,PSEC
      WRITE(LDO,8040)
      NWALK=5
  140 READ(LDI,ERR=9001,END=9998)AF,AS,AC,XAI
    5 WRITE(LDF'IFADD)CORE,CLAD,XNA,DF,FP,DEL,ALFA,
         XLMC,PN1,PDD,PSEC,AC,AF,AS,XAI
      GO TO 9999
    3 CALL DIRECT(1,NAME,4,$9000,$9999)
    2 READ(LDF'IFADD)CORE,CLAD,XNA,DF,FP,DEL,ALFA,
         XLMC,PN1,PDD,PSEC,AC,AF,AS,XAI
      IF(NOPT.EQ.2)GO TO 9999
    4 WRITE(LDO,8001,ERR=9003)CORE,CLAD,XNA,DF,FP,DEL,AF,
         ALFA,XLMC,PN1,PDD,PSEC,AS,AC,XAI
      GO TO 9999
```

```
8000 FORMAT(" ENTER VALUES FOR:  CORE,DF,FP,DEL")
8020 FORMAT(" ENTER VALUES FOR:  ALFA,XLMC,CLAD,XNA")
8030 FORMAT(" ENTER VALUES FOR:  PN1,PDD,PSEC")
8040 FORMAT(" ENTER VALUES FOR:  AF,AS,AC,XAI")
8001 FORMAT(" CORE=",E11.5,T24,"CLAD=",E11.5,T48," XNA=",E1
\C1.5,/,
     "    DF=",E11.5,T24,"  FP=",E11.5,T48," DEL=",E11.5,/,"
\C    AF=",
     E11.5,T24,"ALFA=",E11.5,T48,"XLMC=",E11.5,/,"   PN1=",E
\C11.5,
     T24," PDD=",E11.5,T48,"PSEC=",E11.5,/,"    AS=",E11.5,T
\C24,
     "  AC=",E11.5,T48," XAI=",E11.5,/," *END FILE*")
9000 CALL ERROR(8)
     GO TO 9999
9001 WRITE(LDD,8002)
8002 FORMAT("0**INPUT ERROR.  PLEASE TRY AGAIN.**")
     GO TO (100,100,120,130,140),NWALK
9003 CALL ERROR(17)
9998 RETURN 1
9999 RETURN
     END
```

```
      SUBROUTINE RAMDAT(NOPT,*)
C     THIS SUBROUTINE CALLED BY:  CREATE,LIST,GET,SAVE
C
C     OPTIONS ARE:
C        1 - READ INITIAL DATA AND WRITE TO FILE
C        2 - READ DATA FROM FILE
C        3 - WRITE A FILE LISTING
C        4 - LIST CURRENT DATA
C        5 - WRITE CURRENT DATA TO FILE
C
      COMMON /ERR/NFLAG(10)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /FAMS/CORE,DF,FP,DEL,XLAM,SPEC,ALFA,XLMC,CLAD,X
\CNA,PN1
      ,PDD,PSEC,AC,AF,AS,CN,SN,XL,XAI,XAO,PFI,PTO,
      CONN,SPLC,FIBR,TRA,TER,TOT,BL,GAI
      COMMON /RAMS/HAD,XIL,XIB,XISQ,RES,XM,SRMS,RNB,XMSC,XMA
\C,BETA,R,
  ^^EEE
\SEGMENT EEE NOT FOUND.
```

```
      SUBROUTINE RAMDAT(NOPT,*)
C     THIS SUBROUTINE CALLED BY:  CREATE,LIST,GET,SAVE
C
C     OPTIONS ARE:
C         1 - READ INITIAL DATA AND WRITE TO FILE
C         2 - READ DATA FROM FILE
C         3 - WRITE A FILE LISTING
C         4 - LIST CURRENT DATA
C         5 - WRITE CURRENT DATA TO FILE
C
      COMMON /ERR/NFLAG(10)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /FAMS/CORE,DF,FP,DEL,XLAM,SPEC,ALFA,XLMC,CLAD,X
     &CNA,PN1
     ,PDD,PSEC,AC,AF,AS,CN,SN,XL,XAI,XAD,PFI,PTD,
      CONN,SPLC,FIBR,TRA,TER,TOT,BL,GAI
      COMMON /RAMS/XAD,XIL,XIB,XISO,RES,XM,SRMS,RNB,XMSC,XMA
     &C,BETA,R,
      TYPE,OPT,GAIN,TBER,TSNR,PFD,CHAN,
      GOPT,BER,SNR,BAND
      COMMON NAME,IFADD
      GO TO (1,2,3,4,5),NOPT
C     READ INITIAL DATA FROM CARDS OR TELETYPE
    1 IF(NFLAG(1).NE.1)GO TO 10
      READ(LDI,ERR=9000)TYPE,GAIN,XAD,XIL,XIB,XISO,RES,XM,SR
     &MS,
      RNB,XMSC,XNA,BETA,R
      IF(TYPE.LT.1.0.OR.TYPE.GT.9.0)GO TO 9000
      GO TO 5
C     INITIALIZE ALL RAM VARIABLES TO ZERO
   10 XM=0.0
      SRMS=0.0
      RNB=0.0
      XMSC=0.0
      XMA=0.0
      BETA=0.0
      R=0.0
      WRITE(LDO,8000)
C     ENTER PARAMETERS COMMON TO ALL RAM ANALYSES
      NWALK=1
  100 READ(LDI,ERR=9001,END=9998)TYPE,GAIN,XAD,XIL,XIB,XISO,
     &ORES
      IF(TYPE.LT.1.0.OR.TYPE.GT.9.0)GO TO 9000
      IF(TYPE.GT.1.0)GO TO 120
C
C     ENTER PARAMETERS FOR BASEBAND ANALOG
      WRITE(LDO,8010)
      NWALK=2
  110 READ(LDI,ERR=9001,END=9998)XM,SRMS,RNB
      GO TO 1000
```

```
C
C       ENTER PARAMETERS FOR MULTI-ANALOG
  120 IF(TYPE.NE.2.0)GO TO 140
      WRITE(LDO,8020)
      NWALK=3
  130 READ(LDI,ERR=9001,END=9998)SRMS,RNB,XMSC,XMA
      GO TO 1000
  140 IF(TYPE.NE.3.0)GO TO 160
      WRITE(LDO,8030)
      NWALK=4
  150 READ(LDI,ERR=9001,END=9998)SRMS,RNB,XMSC,BETA
      GO TO 1000
  160 IF(TYPE.NE.4.0.AND.TYPE.NE.5.0)GO TO 180
      WRITE(LDI,8040)
      NWALK=5
  170 READ(LDI,ERR=9001,END=9998)SRMS,RNB,XMSC
      GO TO 1000
  180 IF(TYPE.NE.6.0.AND.TYPE.NE.7.0)GO TO 200
      WRITE(LDO,8050)
      NWALK=6
  190 READ(LDI,ERR=9001,END=9998)RNB,XMSC
      GO TO 1000
  200 WRITE(LDO,8060)
      NWALK=7
  210 READ(LDI,ERR=9001,END=9998)R
 1000 CONTINUE
    5 WRITE(LDF'IFADD)TYPE,GAIN,XAD,XIL,XIB,XISO,RES,
      XM,SRMS,RNB,XMSC,XMA,BETA,R
      GO TO 9999
    3 CALL DIRECT(2,NAME,4,$9998,$9999)
    2 READ(LDF'IFADD)TYPE,GAIN,XAD,XIL,XIB,XISO,RES,XM,SRMS,
\CRNB,
          XMSC,XMA,BETA,R
      IF(NOPT.EQ.2)GO TO 9999
    4 WRITE(LDO,8001,ERR=9003)TYPE,GAIN,XAD,XIL,XIB,XISO,RES
\C,XM,
      SRMS,RNB,XMSC,XMA,BETA,R
      GO TO 9999
 8000 FORMAT("0ENTER VALUES FOR: SRCE,GAIN,XAD,XIL,XIB,XISO,
\CRES")
 8001 FORMAT(T2,"    SRCE=",E11.5,T24,"GAIN=",E11.5,T48,"XAD
\C=",E11.5,
      /,T2,"     XIL =",E11.5,T24,"XIB =",E11.5,T48,"XISO=",E
\C11.5,/,T2,
      "      RES =",E11.5,T24,"XM  =",E11.5,T48,"SRMS=",E11.5,
\C/,T2,
      "      RNB =",E11.5,T24,"XMSC=",E11.5,T48,"XMA =",E11.5,
\C/,T2,
      "      BETA=",E11.5,T24,"R   =",E11.5,/," *END FILE*")
```

```
8010 FORMAT(" ENTER VALUES FOR: XM,SRMS,RNB")
8020 FORMAT(" ENTER VALUES FOR: SRMS,RNB,XMSC,XMA")
8030 FORMAT(" ENTER VALUES FOR: SRMS,RNB,XMSC,BETA")
8040 FORMAT(" ENTER VALUES FOR: SRMS,RNB,XMSC")
8050 FORMAT(" ENTER VALUES FOR: RNB,XMSC")
8060 FORMAT(" ENTER A VALUE FOR: R")
9000 CALL ERROR(21)
     GO TO 9998
9001 WRITE(LIO,8002)
9002 FORMAT("0**INPUT ERROR. PLEASE TRY AGAIN.**")
     GO TO (100,110,130,150,170,190,210),NWALK
9003 CALL ERROR(17)
9998 RETURN 1
9999 RETURN
     END
```

```
      SUBROUTINE SAMDAT(NOPT,NRT)
      DIMENSION XTYPE(30),XXISO(30),XSRMS(30),XXMA(30),XBETA
\C(30)
      COMMON /ERR/NFLAG(10)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /FAMS/CORE,DF,FP,DEL,XLAM,SPEC,ALFA,XLMC,CLAD,X
\CNU,PN1
      ,PDD,PSEC,AC,AF,AS,CN,SN,XL,XAI,XAD,PFI,PTO,
      CONN,SPLC,FIBR,TRA,TER,TOT,BL,GAI
      COMMON /RAMS/XAD,XIL,XIB,XISO,RES,XM,SRMS,RNB,XMSC,XMA
\C,BETA,R,
      TYPE,OPT,GAIN,TBER,TSNR,PFO,CHAN,
       GOPT,BER,SNR,BAND
      COMMON /SPECL/XTYPE,XXISO,XSRMS,XXMA,XBETA,DPR
      COMMON NAME,IFADD
      GO TO (1,2,3,4,5),NOPT
    1 IF(NFLAG(1).NE.1)GO TO 10
      READ(LDI,ERR=9000)CN,SN,XL,XAD,OPT,TBER,TSNR,CHAN
      GO TO 5
   10 WRITE(LDO,8000)
      NWALK=1
  100 READ(LDI,ERR=9001,END=9998)CN,SN,XL,XAD
      WRITE(LDO,8010)
      NWALK=2
  110 READ(LDI,ERR=9001,END=9998)OPT,TBER,TSNR,CHAN
    5 WRITE(LDF'IFADD)CN,SN,XL,XAD,OPT,TBER,TSNR,NC,(XTYPE(I
\C),XSRMS(I),XXISO(I),XXMA(I),XBETA(I),I=1,NC)
      GO TO 9999
    3 CALL DIRECT(4,NAME,4,$9000,$9999)
    2 READ(LDF'IFADD)CN,SN,XL,XAD,OPT,TBER,TSNR,NC,(XTYPE(I)
\C,XSRMS(I),XXISO(I),XXMA(I),XBETA(I),I=1,NC)
      IF(NOPT.EQ.2)GO TO 9999
    4 WRITE(LDO,8001,ERR=9003)CN,SN,XL,XAD,OPT,TBER,TSNR,CHA
\CN
      GO TO 9998
 8000 FORMAT(" ENTER VALUES FOR: CN,SN,XL,XAD")
 8010 FORMAT(" ENTER VALUES FOR: OPT,TBER,TSNR,CHAN")
 8001 FORMAT(" CN  =",E11.5,T24,"SN  =",E11.5,T48,"XL  =",
      E11.5,/," XAD =",E11.5,T24," OPT=",E11.5,T48,"TBER=",
      E11.5,/,"TSNR ="E11.5,T24,"CHAN=",E11.5,/," *END FILE*
\C")
 9000 CALL ERROR(8)
      GO TO 9999
 9001 WRITE(LDO,8002)
 8002 FORMAT("0**INPUT ERROR. PLEASE TRY AGAIN.**")
      GO TO (100,110),NWALK
 9003 CALL ERROR(17)
 9998 RETURN NRT
 9999 RETURN
      END
```

```
      SUBROUTINE CREATE
      INTEGER BLANK
      DIMENSION NA(3)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /ERR/NFLAG(10)
      DATA BLANK/4H    /,D/1HD/
C     GET SECOND WORD OF COMMAND LINE
      NA(1)=BLANK
      NA(2)=BLANK
       NA(3)=BLANK
      CALL DCODE(D,NBEG,NEND,NLEN,$9000)
      IF(NLEN.GT.9)GO TO 9080
      ENCODE(NA,1000)(LINE(K),K=NBEG,NEND)
C     IDENTIFY SECOND WORD
      CALL SECFIL(NA,I,$9010)
      IF(I.NE.5)GO TO 7
C     DEAL NOW ONLY WITH CREATING DIRECTORIES
      CALL DCODE(D,NBEG,NEND,NLEN,$5)
      IF(NLEN.GT.3)GO TO 9020
      NA(1)=BLANK
      ENCODE(NA,1000)(LINE(K),K=NBEG,NEND)
      CALL MODULE(NA(1),I,$9050)
    5 CALL DIRECT(I,NA(1),1,$9999,$9999)
      WRITE(LDO,8000)
      GO TO 9999
C
C     ADD A NEW FILE
    7 CALL DCODE(D,NBEG,NEND,NLEN,$9060)
      IF(NLEN.GT.4)GO TO 9020
      NA(1)=BLANK
      ENCODE(NA,1000)(LINE(K),K=NBEG,NEND)
C
C     TEST FOR UNIQUE NAME
      CALL DIRECT(I,NA(1),3,$9040,$9999)
      CALL DIRECT(I,NA(1),6,$9070,$9999)
      CALL DATCAL(I,1,$9999)
      CALL DIRECT(I,NA(1),2,$9999,$9999)
      WRITE(LDO,8001)NA(1)
      GO TO 9999
```

```
1000 FORMAT(72A1)
8000 FORMAT(1X,"*DIRECTORY CREATED*")
8001 FORMAT(1X,"*FILE ",A4,"CREATED*")
9000 CALL ERROR(1)
     GO TO 9999
9010 CALL ERROR(4)
     GO TO 9999
9020 CALL ERROR(3)
     GO TO 9999
9040 CALL ERROR(7)
     GO TO 9999
9050 CALL ERROR(9)
     GO TO 9999
9060 CALL ERROR(10)
     GO TO 9999
9070 CALL ERROR(11)
     GO TO 9999
9080 CALL ERROR(16)
9999 RETURN
     END
```

```
      SUBROUTINE DELETE
      INTEGER BLANK
      DIMENSION NA(2)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      DATA BLANK/4H    /,O/1HO/
      CALL DCODE(O,NBEG,NEND,NLEN,$9000)
      IF(NLEN.GT.7)GO TO 9040
      NA(1)=BLANK
      NA(2)=BLANK
      ENCODE(NA(1),1000)(LINE(K),K=NBEG,NEND)
      CALL SECFIL(NA,I,$9010)
      IF(I.EQ.5)GO TO 9020
C      GET FILE NAME
      CALL DCODE(O,NBEG,NEND,NLEN,$9000)
      IF(NLEN.GT.4)GO TO 9030
      NA(1)=BLANK
      ENCODE(NA(1),1000)(LINE(K),K=NBEG,NEND)
C
C     LOOK UP FILE IN DIRECTIRY
      CALL DIRECT(I,NA(1),4,$9999,$9999)
C
C     DELETE FILE
      CALL DIRECT(I,NA(1),5,$9999,$9999)
      WRITE(LDO,8000)
      GO TO 9999
 1000 FORMAT(72A1)
 8000 FORMAT(1X,"*FILE DELETED*")
 9000 CALL ERROR(1)
      GO TO 9999
 9010 CALL ERROR(4)
      GO TO 9999
 9020 CALL ERROR(12)
      GO TO 9999
 9030 CALL ERROR(3)
      GO TO 9999
 9040 CALL ERROR(16)
 9999 RETURN
      END
```

```
      SUBROUTINE GET
      INTEGER BLANK
      DIMENSION NA(3),LAB(2),NAM(1)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /ERR/NFLAG(10)
      COMMON NAME,IFADD
      DATA LAB/4HIREC,4HT    /
      DATA BLANK/4H    /,O/1HO/
      NENT=0
      GO TO 1
      ENTRY LIST
      NENT=1
    1 CALL DCODE(O,NBEG,NEND,NLEN,$9000)
      IF(NLEN.GT.9)GO TO 9050
      NA(1)=BLANK
      NA(2)=BLANK
      NA(3)=BLANK
      ENCODE(NA,1000) (LINE(K),K=NBEG,NEND)
      CALL SECFIL(NA,I,$2)
      CALL DCODE(O,NBEG,NEND,NLEN,$4)
      IF(NLEN.GT.4)GO TO 9030
      NAM(1)=BLANK
      ENCODE(NAM,1000) (LINE(K),K=NBEG,NEND)
      CALL DIRECT(I,NAM,4,$9060,$9999)
C     CALL ROUTINES WHICH ACTUALLY HANDLE DATA
      IF(NENT.EQ.0)CALL DATCAL(I,2,$9999)
      IF(NENT.EQ.0)CALL DATCAL(I,3,$9999)
      IF(NENT.EQ.0)WRITE(LDO,8000)NA(1),NA(2),NAM
      NFLAG(I+3)=1
      GO TO 9999
    2 IF(NENT.EQ.0)GO TO 9010
      CALL MODULE(NA(1),L,$9040)
      IF(NA(2).NE.LAB(1).OR.NA(3).NE.LAB(2))GO TO 9040
    3 IF(NFLAG(2).EQ.1)GO TO 9999
      CALL DIRECT(L,NAME,7,$9999,$9999)
      GO TO 9999
    4 CALL DATCAL(I,4,$9999)
                                      GO TO 9999
```

```
1000 FORMAT(72A1)
8000 FORMAT(1X,2A4,1X,A4,1X,"RETRIEVED")
9000 CALL ERROR(1)
      GO TO 9999
9010 CALL ERROR(4)
     GO TO 9999
9020 CALL ERROR(12)
     GO TO 9999
9030 CALL ERROR(3)
     GO TO 9999
9040 CALL ERROR(14)
     GO TO 9999
9050 CALL ERROR(16)
     GO TO 9999
9060 CALL ERROR(15)
9999 RETURN
     END
```

```
      SUBROUTINE SAVE
      INTEGER BLANK
      DIMENSION NA(2)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /ERR/NFLAG(10)
      DATA NYES,NO/4HYES ,4HNO  /,BLANK/4H    /
      DATA D/1HD/
      MFLAG=0
C      IDENTIFY FILE TYPE
      CALL DCODE(D,NBEG,NEND,NLEN,$9000)
      IF(NLEN.GT.7)GO TO 9060
      NA(1)=BLANK
      NA(2)=BLANK
      ENCODE(NA,1000) (LINE(K),K=NBEG,NEND)
      CALL SECFIL(NA,I,$9010)
C     GET THE FILENAME
      CALL DCODE(D,NBEG,NEND,NLEN,$9000)
      IF(NLEN.GT.4)GO TO 9040
      NA(1)=BLANK
      ENCODE(NA(1),1000) (LINE(K),K=NBEG,NEND)
C     GET THE FILE ADDRESS
      CALL DIRECT(I,NA(1),4,$1,$1)
      WRITE(LDO,8000)
    3 READ(LDI,1001)NANS
      IF(NANS.EQ.NO)GO TO 9999
      IF(NANS.EQ.NYES)GO TO 2
      WRITE(LDO,8001)
      GO TO 3
    1 MFLAG=1
      CALL DIRECT(I,NA(1),6,$9050,$9050)
    2 CALL DATCAL(I,5,$9999)
      IF(MFLAG.EQ.1)CALL DIRECT(I,NA(1),2,$9999,$9999)
      WRITE(LDO,8002)
      GO TO 9999
```

```
 1000 FORMAT(72A1)
 1001 FORMAT(A4)
 8000 FORMAT(1X,"****YOU ARE ABOUT TO WRITE OVER AN EXISTING
\C FILE.",/,
      1X,"****ENTER YES TO CONTINUE, NO TO ABORT COMMAND.")
 8001 FORMAT(1X,"PLEASE ANSWER YES OR NO.  NO WILL RETURN YO
\CU TO CONTROL
      MODE.")
 8002 FORMAT(1X,"*FILE SAVED*")
 9000 CALL ERROR(1)
      GO TO 9999
 9010 CALL ERROR(4)
      GO TO 9999
 9020 CALL ERROR(12)
      GO TO 9999
 9040 CALL ERROR(3)
      GO TO 9999
 9050 CALL ERROR(11)
      GO TO 9999
 9060 CALL ERROR(16)
 9999 RETURN
      END
```

```
      SUBROUTINE TAM(*)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /TAMS/TAIL,XLEN,XID,BTR,PTI,SRCE,RT,ANGL,DS,
      BW,TNA,TFP,TDF,RTI
      COMMON /FAMS/CORE,DF,FP,DEL,XLAM,SPEC,ALFA,XLMC,CLAD,X
\CNU,PN1
      ,PDD,PSEC,AC,AF,AS,CN,SN,XL,XAI,XAO,PFI,PTO,
      CONN,SPLC,FIBR,TRA,TER,TOT,BL,GAI
C
C     PTI - POWER INPUT TO TAM ROUTINE
C     PR  - GLOBAL AVAILABLE POWER
C     PTO - POWER OUTPUT TO FIBER CABLE
C
C     SOURCE TYPES               LENS SYSTEM      PIGTAIL
C       1 - EDGE LED               0 - NO           0 - NO
C       2 - SURFACE LED            1 - YES          1 - YES
C       3 - LASER ILD
C       4 - LASER YAG
C
      IF(RT.EQ.0.0.AND.BW.EQ.0.0)GO TO 9000
      RTI=RT
      BWI=BW
      IF(BWI.EQ.0.0)BWI=0.35/RT
      IF(RTI.EQ.0.0)RTI=0.35/BW
    1 PTO=PTI
      PR=PTI
      NSRCE=SRCE
      IF(TAIL.EQ.0.0.AND.(NSRCE.NE.2.OR.PTI.NE.0.0))GO TO 99
\C99
      GO TO (6,2,3,4),NSRCE
    6 PTO=0.1*0.5*PTI*TNA**2*TDF**2*TFP/(0.14*90.0)**2
      GO TO 23
    2 IF(PTI.EQ.0.0.AND.BTR.EQ.0.0.AND.XID.EQ.0.0)GO TO 9000
      IF(ANGL.LE.0.0.OR.ANGL.GT.90.0)GO TO 9010
      FN=90.0/ANGL
      A=1.0
      IF(FN.NE.1.0)A=COS(3.1416*0.5*(FN-1.0)/FN)/(FN-1.0)-CO
\CS(3.1416*
      0.5*(FN+1.0)/FN)/(FN+1.0)-2.0/(FN**2-1.0)
      PS=PTI
      IF(PTI.NE.0.0)GO TO 5
      IF(BTR.NE.0.0)PS=3.1416**2*A*BTR*DS**2/4.0E8
      IF(XID.NE.0.0)PS=3.1416*A*XID
    5 PR=PS
      PTO=PS
      IF(TAIL.EQ.0.0)GO TO 9999
      C=TNA**2
      IF(FN.NE.1.0)C=COS(TNA*(FN-1.0))/(FN-1.0)-COS(TNA*(FN+
```

```
\C1.0)+/
     (FN+1.0)-2.0/(FN**2-1.0)
     D=1.0
     IF(TDF.LT.DS)D=TDF**2/DS**2
     PTO=PS*C*TFP*D/A
     GO TO 23
   3 IF(XLEN.EQ.0.0)PTO=PTI*0.15*TFP
     IF(XLEN.NE.0.0)PTO=PTI*0.5*TFP
     GO TO 23
   4 RAT=TDF**2/DS**2
     IF(RAT.GT.1.0)RAT=1.0
     PTO=TFP*PTI*RAT
  23 PWR=PTI
     IF(PTI.EQ.0.0)PWR=PS
     GAI=XAI
     IF(GAI.EQ.0.0.AND.TNA.LT.ANGL)GAI=10.0*ALOG10(PWR/PTO)
9999 WRITE(LDO,100)SRCE,TAIL,BWI,PTI,PR,PTO
 100 FORMAT("0*TAM*",/,"  SRCE=",E15.9,T24,"TAIL=",E15.9,T4
\C8,"  BW=",
     E15.9,/,"   PTI=",E15.9,T24,"  PR=",E15.9,T48," PTO=",
\CE15.9)
     WRITE(LDO,200)XIO,BTR,XLAM,SPEC,TNA,ANGL
 200 FORMAT("   XIO=",E15.9,T24," BTR=",E15.9,T48,"XLAM=",E
\C15.9,/,
     "  SPEC=",E15.9,T24,"TNA=",E15.9,T48,"ANGL=",E15.9)
     RETURN
9000 CALL ERROR(22)
     GO TO 9998
9010 CALL ERROR(24)
9998 RETURN 1
     END
```

```
      SUBROUTINE FAM(*)
C
C   INPUT IS EITHER CORE AND CLADDING REFRACTIVE INDICES OR T
\CHE NUMERICAL
C   APERTURE AND FRACTIONAL INDEX DIFFERENCE
C
      COMMON /TAMS/TAIL,XLEN,XID,BTR,PTI,SRCE,RT,ANGL, DS,BW
\C,TNA,
      TFP,TDF,RTI
      COMMON /FAMS/CORE,DF,FP,DEL,XLAM,SPEC,ALPHAF,XLMC,CLAD
\C,XNU,PN1,
      PDD,PSEC,AC,AF,AS,CN,SN,XL,XAI,XAO,PFI,PTO,
      CONN,SPLC,FIBR,TRA,TER,TOT,BL,GAI
      COMMON /PAMS/XAD,XIL,XIB,XISO,RES,XM,SRMS,PNB,XMSC,XMA
\C,BETA,R,
      TYPE,OPT,GAIN,TBER,TSNR,PFD,CHAN,
      GOPT,BER,SNR,BAND
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
C
C   THE NEXT 10 VARIABLES MAY HAVE THEIR VALUES CHANGED WITHI
\CN THE SUBROUTINE.
C   THEY HAVE THEREFORE BEEN MADE ONE-WAY VARIABLES.
C
      IF(PTO.EQ.0.0.AND.PFI.EQ.0.0)GO TO 10000
      PCI=PFI
      IF(PFI.EQ.0.0)PCI=PTO
      COREI=CORE
      DELTA=DEL
      CLADI=CLAD
      XNA=XNU
      PNA=PN1
      PND=PDD
      PNS=PSEC
C
C   SET UP DEFAULT GLASS PARAMETERS
C
      IF(PNA.EQ.0.0)PNA=-0.0145
      IF(PND.EQ.0.0)PND=-0.00085
      IF(PNS.EQ.0.0)PNS=0.025
      IF(XLAM.EQ.0.0)GO TO 10000
      IF(DELTA.GE.1.0)GO TO 9000
      IF(COREI.EQ.0.0.AND.DELTA*XNA.NE.0.0)COREI=XNA/SQRT(2.
\C0*DELTA)
      IF(CLADI.EQ.0.0.AND.DELTA*COREI.NE.0.0)CLADI=COREI*(1.
\C0-DELTA)
      IF(DELTA.EQ.0.0.AND.COREI*CLADI.NE.0.0)DELTA=(COREI-CL
\CADI)/COREI
      IF(XNA.EQ.0.0.AND.COREI*DELTA.NE.0.0)XNA=COREI*SQRT(2.
\C0*DELTA)
C
C   CHECK THESE VARIABLES TO BE CERTAIN THEY HAVE BEEN DEFINE
\CD
C
```

```
      IF(COREI*DELTA.EQ.0.0)GO TO 10000
C
C     CALCULATE POWER RECEIVED BY FIBER CABLE
      IF(PTO.EQ.0.0.OR.TAIL.EQ.1.0)GO TO 21
C
C     PTO HAS BEEN DEFINED AND WE HAVE NO PIGTAIL
      IF(SRCE.EQ.1.0)PCI=0.1*(XNA*DF)**2*FP*PTO/(0.14*90.0)*
\C*2*0.5
      IF(SRCE.EQ.3.0.AND.XLEN.EQ.0.0)PCI=0.15*FP*PTO
      IF(SRCE.EQ.3.0.AND.XLEN.NE.0.0)PCI=0.5*FP*PTO
      IF(SRCE.EQ.4.0)PCI=FP*PTO*DF**2/DS**2
      IF(SRCE.NE.2.0)GO TO 23
      IF(ANGL.LE.0.0.OR.ANGL.GT.90.0)GO TO 10000
      FN=90.0/ANGL
      A=1.0
      C=XNA**2
      IF(FN.EQ.1.0)GO TO 22
      A=COS(3.1416*0.5*(FN-1.0)/FN)/(FN-1.0)-COS(3.1416*0.5*
      (FN+1.0)/FN)/(FN+1.0)-2.0/(FN**2-1.0)
      C=COS(TNA*(FN-1.0))/(FN-1.0)-COS(TNA*(FN+1.0))/(FN+1.0
\C)-2.0/
      (FN**2-1.0)
   22 D=1.0
      IF(DF/DS.LE.1.0)D=DF**2/DS**2
      PCI=PTO*C/A*FP*D
C
C     COMPUTE INPUT COUPLING LOSS
   23 GAI=XAI
      IF(GAI.EQ.0.0.AND.XNA.LT.ANGL)GAI=10.0*ALOG10(PTO/PCI)
C
C  COMPUTE INTERMEDIATE GLASS PARAMETERS
C
   21 CN1=COREI-PNA
      EPS=-2.0*COREI*PND/(CN1*DELTA)
      C2=(3.0*ALPHAF-2.0*EPS-2.0)/(2.0*ALPHAF+4.0)
      C1=(ALPHAF-EPS-2.0)/(ALPHAF+2.0)
C
C  TABLE LOOKUP - INTERPOLATION FOR K(ALPHA) GIVEN ALPHA.
C
      CALL ALFA(ALPHAF,XKAF,$10000)
      XLC=0.0
      IF(XLMC.NE.0.0)XLC=XKAF/XLMC
C
C  COMPUTE INTERMODAL DISPERSION - USE 2 STEPS
C
   11 DTER=XL*CN1*DELTA*ALPHAF/(6.0E-4*(ALPHAF+1.0))*SQRT((A
\CLPHAF
      +2.0)/(3.0*ALPHAF+2.0))
      DTER=DTER*SQRT(C1**2+4.0*C1*C2*DELTA*(ALPHAF+1.0)/(2.0
\C*ALPHAF
      +1.0)+4.0*DELTA**2*C2**2*(2.0*ALPHAF+2.0)**2/((5.0*ALP
\CHAF+2.0)*
      (3.0*ALPHAF+2.)))
```

```
C
C   COMPUTE INTRAMODIAL DISPERSION
C
      DTRA=XL*SPEC/(3.0E-4*XLAM)*SQRT(PNS**2-2.0*
      PNS*CN1*DELTA*((ALPHAF-2.0-EPS)/(ALPHAF+2.0))*(2.0*
      ALPHAF/(2.0*ALPHAF+2.0))+(CN1*DELTA)**2*((ALPHAF-2.0-E
\CPS)/(ALPHAF
      +2.0))**2*((2.0*ALPHAF)/(3.0*ALPHAF+2.0)))
C
C   COMPUTE TOTAL DISPERSION
C
      DTOT=DTER**2
      IF(XL.GE.XLC.AND.XLC.NE.0.0)DTOT=DTOT*XLC/XL
      DTOT=SQRT(DTOT+DTRA**2)
C
C   COMPUTE 3 DB FIBER BANDWIDTH
C
      BL=0.132/(DTOT*1.0E-9)
      CONN=CN*AC
      SPLC=SN*AS
      FIBR=XL*AF
      XLS=CONN+SPLC+FIBR+XAD
      PFD=PCI*10.0**(-0.1*XLS)
      WRITE(LDO,100)DTRA,DTER,DTOT,BL,XLC,PTO,PFI,PFD
  100 FORMAT("0*FAM*",/,"   DTRA=",E15.9,T24,"DTER=",E15.9,T4
\C8,"DTOT=",
      E15.9,/,"    BL=",E15.9,T24," XLC=",E15.9,T48," PTO=",E
\C15.9,/,
      "    PFI=",E15.9,T24," PFD=",E15.9)
      WRITE(LDO,200)XLAM,SPEC,ALPHAF,DEL,XNA,COREI,CLADI,GAI
\C,XLS
  200 FORMAT("  XLAM=",E15.9,T24,"SPEC=",E15.9,T48,"ALFA=",E
\C15.9,/,
      "    DEL=",E15.9,T24," XNA=",E15.9,T48,"CORE=",E15.9,/,
\C"  CLAD=",
      E15,9,T24," XAI=",E15.9,T48,"LOSS=",E15.9)
      RETURN
C
C  THE RIGHT COMBINATION OF INPUT DATA (COREI,CLADI,XNA,DELT
\CA) WAS NOT GIVEN
C
10000 WRITE(LDO,10001)
10001 FORMAT(" DATA INSUFFICIENT FOR FURTHER FIBER ANALYSIS.
\C")
      RETURN 1
 9000 CALL ERROR(24)
      RETURN 1
      END
```

```
      SUBROUTINE ALFA(A,ALPHA,NPT)
      REAL KAL
      DIMENSION ALF(60),KAL(60)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      DATA SAVE/0.0/
      DATA ALF/1.43,1.50,1.55,1.60,1.65,1.7,1.75,1.8,1.85,1.
\C9,1.95,
      2.0,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3.0,3.1,3.2,3.
\C3,3.4,3.5,
      3.6,3.7,3.8,3.9,4.0,4.5,5.0,5.5,6.0,6.5,7.0,7.5,8.0,8.
\C5,9.0,9.5,
      10.0,12.0,14.0,16.0,18.0,20.0,25.0,30.0,35.0,40.0,4=.0
\C,50.0,60.0,
      70.0,80.0,90.0,100.0/
      DATA KAL/3.5,3.2,3.02,2.77,2.73,2.62,2.5,2.42,2.33,2.2
\C5,2.17,2.10,
      1.95,1.87,1.77,1.69,1.62,1.55,1.49,1.44,1.39,1.35,1.32
\C,1.28,1.25,
      1.23,1.2,1.18,1.16,1.14,1.13,1.1,1.05,0.99,0.95,0.92,0
\C.89,0.86,
      0.84,0.82,0.80,0.79,0.78,0.77,0.73,0.7,0.67,0.66,0.64,
\C0.62,0.6,
      0.59,0.58,0.57,0.57,0.56,0.56,0.56,0.55,0.55/
      IF(A.EQ.SAVE)RETURN
      IF(A.LT.1.43.OR.A.GT.100.0)GO TO 20
      DO 10 I=1,60
      IF(ALF(I).EQ.A)GO TO 40
      IF(ALF(I).GT.A)GO TO 50
   10 CONTINUE
   20 WRITE(LDO,100)
  100 FORMAT("0ALPHA MUST BE WITHIN THE RANGE OF 1.43 TO 100
\C.0 INCLUSIVE
      .")
      RETURN NRT
   40 ALPHA=KAL(I)
      RETURN
   50 ALPHA=KAL(I)-(ALF(I)-A)/(ALF(I)-ALF(I-1))*(KAL(I)-KAL(
\CI-1))
      SAVE=ALPHA
      RETURN
      END
```

```
      SUBROUTINE RAM(*)
      DIMENSION TRAN(30,5),P(2,35)
      COMMON /RAMS/XAD,XIL,XIB,XISO,RES,XM,SRMS,RNB,XMSC,XMA
\C,BETA,R,
      TYPE,OPT,GAIN,TBER,TSNR,PFD,CHAN,
      GOPT,BER,SNR,BAND
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
      COMMON /FAMS/CORE,DF,FP,DEL,XLAM,SPEC,ALFA,XLMC,CLAD,X
\CNU,PN1,
      PDD,PSEC,AC,AF,AS,CN,SN,XL,XAI,XAD,PFI,PTD,
      CONN,SPLC,FIBR,TRA,TER,TDT,BL,GAI
      COMMON /SPECL/TRAN,DPR
      DATA P/0.10383,1.26,0.01044,2.31,0.10349E-2,3.08,0.103
\C71E-3,
      3.71,0.13348E-4,4.2,0.54129,4.39,0.79335E-6,4.799,0.47
\C919E-6,
      4.899,0.28665E-6,4.999,0.16983E-6,5.099,0.99645E-7,5.1
\C99,
      0.57902E-7,5.299,0.33321E-7,5.399,0.10718E-7,5.599,0.5
\C9904E-8,
      5.699,0.33158E-8,5.799,0.18175E-8,5.899,0.98659E-9,5.9
\C99,
      0.53034E-9,6.099,0.28232E-9,6.199,0.14982E-9,6.299,0.7
\C7689E-10,
      6.399,0.40160E-10,6.499,0.20558E-10,6.599,0.10421E-10,
\C6.699,
      0.12798E-11,6.999,0.14388E-12,7.299,0.14807,7.599,0.13
\C945E-14,
      7.999,0.12019E-1=,8.19,0.22224E-16,8.399,0.16594E-17,8
\C.699,
      0.11286E-18,8.999,0.17898E-19,9.199,0.10495E-20,9.49/
C◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
\C◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
C
C     TYPE          EXPLANATION
C      1       ANALOG - BASEBAND
C      2       ANALOG - AM
C      3       ANALOG - FM
C      4       ANALOG - DSB
C      5       ANALOG - SSB
C      6       DIGITAL- FSK
C      7       DIGITAL- PSK
C      8       DIGITAL- BASEBAND (OOK)
C      9       DIGITAL- BASEBAND (BPPM)
```

```
C
      INDX=0
      PR=PFD*1.0E9
      IF(PFD.EQ.0.0)GO TO 10070
      IF(CHAN.LE.1.0)GO TO 17
   19 INDX=INDX+1
      IF(FLOAT(INDX).GT.CHAN)GO TO 9999
      NTYPE=TRAN(INDX,1)
      XISQA=TRAN(INDX,2)**2
      SRMS=TRAN(INDX,3)
      XMA=TRAN(INDX,4)
      BETA=TRAN(INDX,5)
      GO TO 18
   17 NTYPE=TYPE
      XISQA=XISQ**2
   18 XGAIN=GAIN
      GOPT=0.0
      NGOPT=OPT
      IF(GAIN.LE.0.0)XGAIN=1.0
      SNB=RNB
      BAND=RNB
      IF(NTYPE-8)10030,10010,10020
C
C  DEFINES SNB FOR 2 CASES - DIGITAL BASEBAND OOK AND BPPM
C
10010 SNB=R*3.1416/3.0
      BAND=R/2.0
      GO TO 10030
10020 SNB=R*3.1416*2.0/3.0
      BAND=R
C
C  SHOULD WE OPTIMIZE GAIN?
C
10030 IF(NGOPT.EQ.1.AND.NTYPE.LT.6)XGAIN=((2.0*1.6021E-10*XI
\IL*SNB+XISQA
      )/(1.6021E-10*SNB*XAD*(RES*PR+XIB)))**(1.0/(2.0+XAD))
C
C  EVALUATE SOME INTERMEDIATE EXPRESSIONS COMMON TO ALL ANAL
\CYSES
C
      XDSQA=4.0*1.6021E-10*RES*PR*SNB*XGAIN**(2.0+XAD)
      IF(NTYPE.EQ.8.OR.NTYPE.EQ.9)SNB=SNB/2.0
      XDSQA=2.0*1.602E-10*SNB*(XIL+XIB*XGAIN**(2.0+XAD))
      XINSQ=XDSQA+XDSQA+XISQA
C
C  IF WE ARE DOING AN ANALOG ANALYSIS, GO TO 10060
C
      IF(NTYPE.LT.8)GO TO 10040
C
C  CALCULATE DIGITAL BASEBAND (OOK,BPPM)
```

```
C
      Q=SQRT(2.0)*4.0/3.1416*RES*XGAIN*PR/(SQRT(XINSQ)+SQRT(
     CXDSQA+XISQA)
     )
      SNB=BAND
      GO TO 10050
C
C  COMPUTE EB/N0
C
10040 IF(NTYPE.LT.6)GO TO 10060
      ECR=0.5*RES**2*XGAIN**2*PR**2*XMSC**2/XINSQ
C
C  CALCULATE DIGITAL FSK
C
      IF(NTYPE.EQ.6)Q=1.1*SQRT(ECR)
C
C  CALCULATE DIGITAL PSK
C
      IF(NTYPE.EQ.7)Q=SQRT(2.0*ECR)
C
C  COMPUTE BIT ERROR RATE
C
10050 BER=QNORM(Q)
      IF(TBER.EQ.0.0.OR.BER.EQ.TBER)GO TO 10055
      DO 10051 JK=1,35
      IF(TBER.MQ.P(1,JK))GO TO 10052
      IF(TBER.GT.P(1,JK))GO TO 10053
10051 CONTINUE
      CQ=1.0
      GO TO 10057
10052 CQ=P(2,JK)
      GO TO 10057
10053 JKK=JK-1
C      DPR IS IN WATTS
      CQ=P(2,JKK)-(P(1,JKK)-TBER)/(P(1,JKK)-P(1,JK))*(P(2,JK
     CK)-P(2,JK))
10057 IF(NTYPE.GE.8)DPR=CQ*3.1416*1.0E-9*(SQRT(XINSQ)+SQRT(X
     CDSQA+XISQA))
     /(4.0*SQRT(2.0)*RES*XGAIN)
      IF(NTYPE.EQ.6)DPR=CQ*SQRT(2.0*XINSQ*1.0E-9)/(1.1*RES*X
     CGAIN*XMSC)
      IF(NTYPE.EQ.7)DPR=CQ*SQRT(XINSQ*1.0E-9)/(RES*XGAIN*XMS
     CC)
10055 XMAR=10.0*ALOG10(PR/(DPR*1.0E9))
      IF(INDX.GT.1)GO TO 10056
      WRITE(LDO,2002)XAD,XIB,XIL,RES,RNB,XMSC,R,BAND,TBER,PF
     CD
 2002 FORMAT('0*RAM*",//," XAD =",E15.9,T24,"XIB =",E15.9,T4
     C8,"XIL =",
      E15.9,/," RES =",E15.9,T24,'RNB =",E15.9,T48,"XMSC=",E
     C15.9,/,
      " R    =",E15.9,T24,"BAND=",E15.9,T48,'TBER=",E15.9,/,"
     C PFD =",
      E15.9)
```

```
10056 WRITE(LDO,2003)CHAN,TYPE,GAIN,XISQ,DPR,XMAP,BER
 2003 FORMAT(" CHAN=",E15.9,T24,"TYPE=",E15.9,T48,"GAIN=",E1
\C5.9,/,
      " XISQ=",E15.9,T24,"DPR =",E15.9,T48,"XMAR=",E15.9,/,"
\C BER =",
      E15.9)
      RETURN
C
C  EVALUATE THIS EXPRESSION NOW - IT IS COMMON TO ALL 5 ANAL
\COG MODULATIONS
C
10060 PART1=SRMS**2*RES**2*XGAIN**2*PR**2
C
C  CALCULATE ANALOG BASEBAND
C
      IF(NTYPE.EQ.1)ARG=XM**2*PART1/XINSQ
C
C  CALCULATE ANALOG AM
C
      IF(NTYPE.EQ.2)ARG=(XMSC*XMA)**2*PART1*0.=/XINSQ
C
C  CALCULATE ANALOG FM
C
      IF(NTYPE.EQ.3)ARG=1.5*BETA**2*XMSC**2*PART1/XINSQ
C
C  CALCULATE ANALOG DSB
C
      IF(NTYPE.EQ.4)ARG=0.5*XMSC**2*PART1/XINSQ
C
C  CALCULATE ANALOG SSB
C
      IF(NTYPE.EQ.5)ARG=XMSC**2*PART1/XINSQ
C
C  COMPUTE SIGNAL/NOISE RATIO
C
      SNR=(10.0*ALOG10(ARG))
      IF(OPT.EQ.1.0)GOPT=XGAIN
      IF(TSNR.EQ.0.0.OR.TSNR.EQ.SNR)GO TO 10061
      XARG=10.0**(TSNR/10.0)
      IF(NTYPE.EQ.1)YARG=XARG*XINSQ/XM**2
      IF(NTYPE.EQ.2)YARG=2.0*XARG*XINSQ/(XMSC*XMA)**2
      IF(NTYPE.EQ.3)YARG=XARG*XINSQ/(1.5*BETA**2*XMSC**2)
      IF(NTYPE.EQ.4)YARG=2.0*XARG*XINSQ/XMSC**2
      DPRNTPPE(EAP5)YARBMSARG**I080MXMSC**2
      XMAR=10.0*ALOG10(PR/(DPR*1.0E9))
10061 IF(INDX.GT.1)GO TO 10062
      WRITE(LDO,2004)XAD,XIB,XIL,RES,RNB,XMSC,XM,OPT,BAND,TS
\CNR,PFO
 2004 FORMAT("0*RAM*",//," XAD =",E15.9,T24,"XIB =",E15.9,T4
\C8,"XIL =",
      E15.9,/," RES =",E15.9,T24,"RNB =",E15.9,T48,"XMSC=",E
```

```
\C15.9,
      /," XM   =",E15.9,T24,"OPT =",E15.9,T48,"BAND=",E15.9,/
\C," TSNR=",
      E15.9,T24,"PFD =",E15.9)
10062 WRITE(LDD,2005)CHAN,TYPE,GAIN,GOPT,XISO,SRMS,XMA,BETA,
\CDPR,XMAR,
      SNR
 2005 FORMAT(" CHAN=",E15.9,T24,"TYPE=",E15.9,T48,"GAIN=",E1
\C5.9,/,
      " GOPT=",E15.9,T24,"XISO=",E15.9,T48,"SRMS=",E15.9,/,"
\C XMA =",
      E15.9,T24,"BETA=",E15.9,T48,"DPR =",E15.9,/," XMAR=",E
\C15.9,
      T24,"SNR =",E15.9)
      IF(CHAN.GT.1.0)GO TO 19
 9999 RETURN
10070 WRITE(LDD,10080)
10080 FORMAT(" POWER INPUT IS ZERO.")
      RETURN 1
      END
```

```
      FUNCTION QNORM(X)
      COMMON /INPUT/LINE(72),LDI,LDO,LDF,LDA
C  COMPUTES Y=1-P(X)=Q(X)=PROBABILITY THAT THE RANDOM VARIAB
\CLE
C  EQUAL TO OR GREATER THAN X, WHERE THE RANDOM VARIABLE IS
\CNORMALLY
C  DISTRIBUTED WITH ZERO MEAN.
      AX=ABS(X)
      IF(AX.GT.3.6)GO TO 3
      IF(AX.GT.0.000001)GO TO 1
      QNORM=0.5
      RETURN
    1 D1=0.0498673
      D2=0.0211410
      D3=3.2776E-03
      D4=3.80E-05
      D5=4.889E-05
      D6=5.3E-06
      X2=AX*AX
      X3=X2*AX
      X4=X3*AX
      X5=X4*AX
      X6=X5*AX
      P1=0.5*(1.0+(D1*AX)+(D2*X2)+(D3*X3)+(D4*X4)+(D5*X5)+(D
\C6*X6))**
      (-16)
      QNORM=P1
C  THE MAXIMUM ERROR IS 1.5E-07
      GO TO 4
    3 PI=3.1415927
      IF(AX.GT.10.)GO TO 8
      XP=EXP(-AX**2/2.)/AX/SQRT(2.*PI)
      X2=1./AX**2
      X3=3./AX**4
      X4=15./AX**6
      X5=105./AX**8
      X6=945./AX**10
      X7=10395./AX**12
      P1=XP*(1.0-X2+X3-X4+X5-X6+X7)
      QNORM=P1
C  THE MAXIMUM ERROR IS 150.*EXP(-ABS(X)**2/2.)/ABS(X)**11
    4 IF(X)6,7,7
    6 QNORM=1.0-P1
    7 GO TO 10
    8 QNORM=0.0
      IF(X.LT.0.)QNORM=1.0
C     WRITE(LDO,9)
C   9 FORMAT(2X,"***** Q(X) HAS BEEN SET TO ZERO, X IS GREAT
\CER THAN 10")
   10 RETURN
      END
```

3.14 Sample Programs

The following pages are computer listings of design problems actually run with FODAP. Command lines input by the user are indicated by asterisks.

```
**FODAP WAS DEVELOPED FEBRUARY, 1977 UNDER
THE SPONSORSHIP OF ROME AIR DEVELOPMENT CENTER BY
HARRIS CORPORATION, ESSD, MELBOURNE, FLA.

  READY FOR INPUT
```

```
* CREATE NEWFILE
  ****YOU ARE ABOUT TO DESTROY ANY EXISTING FILES.
  ****ENTER YES TO CONTINUE, NO TO ABORT COMMAND.
* YES
  *DIRECTORY CREATED*
* CREATE TAMFILE TEST
  ENTER VALUES FOR: SRCE,TAIL,XLEN,PTI
  2.0,1.0,0.0,1.0
  ENTER VALUES FOR: XLAM,SPEC,RT,RW
  850.0,15.0,1.0E-8,0.0
  ENTER VALUES FOR: TNA,TFP,TDF
  0.14,0.35,0.023
  ENDER VALUES FOR: ANGL,DS,XIO,BTR
  20.0,0.08,0.0,0.0
  *FILE TEST  CREATED*
* CREATE RAMFILE TESP

  ENTER VALUES FOR: RECEIVER-TYPE,GAIN,XAD,XIL,XIB,XIS0,RES
  8.0,1.0,0.3,100.0,0.03,541.7,0.55
  ENTER A VALUE FOR: R
  1.0E7
  *FILE TESP  CREATED*
* CREATE FAMFILE TEST
  ENTER VALUES FOR:CORE,DF,FP,DFL
  1.4612,85.0,0.35,0.00595
  ENTER VALUES FOR:ALFA,XLMC,CLAD,XNA
  25.0,2.0,1.4525,0.16
  ENTER VALUES FOR:PN1,PDD,PSEC
  0.0,0.0,0.0
  ENTER VALUES FOR:AF,AS,AC,XAI
  10.0,1.0,2.0,0.0
  *FILE TEST  CREATED*
* CREATE SAMFILE TEST
  ENTER VALUES FOR:CN,SN,XL
  2.0,0.0,1.0
  ENTER VALUES FOR:OPT,TREE,TSNR,XA0
  0.0,1.0E-8,2.0
  2.0
  *FILE TEST  CREATED*
* DEFINE TSNR=0.0
* DEFINE XA0=2.0
* SAVE SAMFILE TEST
  ****YOU ARE ABOUT TO WRITE OVER AN EXISTING FILE.
  ****ENTER YES TO CONTINUE, NO TO ABORT COMMAND.
* YES
  *FILE SAVED*
* LIST SAMFILE TEST
    CN= .20000E+01          SN= .00000E+01             XL= .10000E+01
   OPT= .00000E+01        TREE= .10000E-07           TSNR= .00000E+01
   XA0= .20000E+01
  *END FILE*
```

```
*GET RAMFILE TEST
 ERROR**FILE-NAME DOES NOT EXIST**
*GET RAMFILE TESP
 RAMFILE   TESP   RETRIEVED
*SAVE RAMFILE TEST
 *FILE SAVED*
*DELETE RAMFILE TESP
 *FILE DELETED*
*LIST RAMDIRECT
 TEST
*TAM

 *TAM*
  SRCE= .200000000E+01 TAIL= .100000000E+01      BW= .349999920E+08
   PTI= .100000000E+01   PR= .100000000E+01     PTO= .106469244E-01
   XIO= .000000000E+01  BTR= .000000000E+01    XLAM= .850000000E+03
  SPEC= .150000000E+02  TNA= .139999986E+00    ANGL= .200000000E+02
*FAM

 *FAM*
  DTRA= .102775669E+01 DTER= .707843304E+01    DTOT= .407290840E+01
    BL= .324092800E+08  XLC= .399999943E+00     PTO= .106469244E-01
   PFI= .000000000E+01  PFO= .267438532E-03
  XLAM= .850000000E+03 SPEC= .150000000E+02    ALFA= .100117417E-06
   DEL= .594999921E-02  XNA= .159999996E+00    CORE= .146120000E+01
  CLAD= .145249987E+01  XAI= .197277527E+02    LOSS= .160000000E+02
*RAM

 *RAM*
  TYPE= .800000000E+01    R= .100000000E+08    GAIN= .100000000E+01
  BAND= .500000000E+07  PFO= .267438532E-03     BER= .000000000E+01
   XAD= .299999952E+00 XISQ= .541699951E+03     XIL= .100000000E+03
   XIB= .299999975E-01  RES= .549999952E+00     RNB= .000000000E+01
*SAM

 ******

  AVAILABLE POWER (SOURCE) = .26744E-03

        INPUT COUPLING LOSS = .19728E+02
                 FIBER LOSS = .10000E+02
                SPLICE LOSS = .00000E+01
             CONNECTOR LOSS = .40000E+01
       OUTPUT COUPLING LOSS = .20000E+01

     RECEIVER POWER (INPUT) = .26744E-03

                      BER =      .00000E+01

              DESIRED BER =      .10000E-07

        SYSTEM BANDWIDTH = .48930E+07

 ******
```

```
*CREATE TAMFILE TESR
 ENTER VALUES FOR: SRCE,TAIL,XLEV,PTI
*2.0,1.0,0.0,0.0
 ENTER VALUES FOR: XLAM,SPEC,RT,BW
*850.0,15.0,0.15E-7
*0.0
 ENTER VALUES FOR: TNA,TEP,TDF
*0.16,1.0,850.0
 ENDER VALUES FOR: ANGL,DS,XIO,BTR
*90.0,75.0,0.0,66.0
 *FILE TESR  CREATED*
*TAM

 *TAM*
  SRCE= .200000000E+01 TAIL= .100000000E+01     BW= .233333320E+08
   PTI= .000000000E+01   PR= .916026533E-02    PTO= .234502746E-03
   XIO= .000000000E+01  BTR= .660000000E+02   XLAM= .850000000E+03
  SPEC= .150000000E+02  TNA= .159999996E+00   ANGL= .900000000E+02
*ORDER FAM,RAM,SAM
*SYS

 *FAM*
  DTRA= .102775669E+01 DTER= .707843304E+01   DTOT= .407299840E+01
    BL= .324092800E+08  XLC= .309999943E+00    PTO= .234502746E-03
   PFI= .000000000E+01  PFO= .589043975E-05
  XLAM= .850000000E+03 SPEC= .150000000E+02   ALFA= .100117417E-06
   DEL= .594999921E-02  XNA= .159999996E+00   CORE= .146120000E+01
  CLAD= .145249987E+01  XAI= .159175987E+02   LOSS= .160000000E+02

 *RAM*
  TYPE= .800000000E+01    R= .100000000E+08   GAIN= .800000000E+02
  BAND= .500000000E+07  PFO= .589043975E-05    BER= .000000000E+01
   XAD= .299999952E+00 XISA= .541699951E+03    XIL= .100000000E+03
   XIB= .299999975E-01  RES= .549999952E+00    RNB= .000000000E+01

 ******

  AVAILABLE POWER (SOURCE) = .58904E-05

        INPUT COUPLING LOSS = .15918E+02
                 FIBER LOSS = .10000E+02
                SPLICE LOSS = .20000E+01
             CONNECTOR LOSS = .40000E+01
       OUTPUT COUPLING LOSS = .20000E+01

    RECEIVER POWER (INPUT) = .58904E-05

                      BER =      .00000E+01

              DESIRED BER =      .10000E-07

        SYSTEM BANDWIDTH = .48343E+07

 ******
```

```
* DEFINE AVGL=90.0
* SYS

 *TAM*
  SRCE= .200000000E+01 TAIL= .100000000E+01       BW= .233333327E+08
   PTI= .000000000E+01   PR= .916026905E-02      PTO= .234502862E-03
   XIO= .000000000E+01  BTR= .660000000E+02     XLAM= .850000000E+03
  SPEC= .150000000E+02  TVA= .159999996E+00     AVGL= .900000000E+02

 *FAM*
  DTRA= .102775598E+01 DTER= .707843208E+01     DTOT= .407290745E+01
    BL= .324092800E+08  XLC= .309999943E+00      PTO= .234502862E-03
   PFI= .000000000E+01  PFO= .589044521E-05
  XLAM= .850000000E+03 SPEC= .150000000E+02     ALFA= .247879939E-09
   DEL= .594999921E-02  XVA= .159999996E+00     CORE= .146120000E+01
  CLAD= .145249987E+01  XAI= .159176006E+02     LOSS= .100000000E+02

 *RAM*
  TYPE= .800000000E+01    R= .100000000E+08     GAIN= .100000000E+01
  BAND= .500000000E+07  PFO= .589044521E-05      BER= .363320680E-07
   XAD= .299999952E+00 XISQ= .541699951E+03      XIL= .100000000E+03
   XIB= .299999975E-01  RES= .549999952E+00      RVB= .000000000E+01

 ******

  AVAILABLE POWER (SOURCE) = .58904E-05

        INPUT COUPLING LOSS = .15918E+02
                 FIBER LOSS = .10000E+02
                SPLICE LOSS = .00000E+01
             CONNECTOR LOSS = .40000E+01
       OUTPUT COUPLING LOSS = .20000E+01

     RECEIVER POWER (INPUT) = .58904E-05

                        BER =     .36332E-07

                DESIRED BER =     .10000E-09

          SYSTEM BANDWIDTH = .48343E+07

 ******

* DEFINE AVGL=89.0
* TAM
```

```
*TAM*
 SRCE= .200000000E+01 TAIL= .100000000E+01       BW= .233333320E+08
  PTI= .000000000E+01   PR= .819690794E+00      PTO= .810943604E+00
  XIO= .000000000E+01  BTR= .660000000E+02     XLAM= .850000000E+03
 SPEC= .150000000E+02  TVA= .159999996E+00     AVGL= .890000000E+02
* DEFINE AVGL=60.0
* TAM

*TAM*
 SRCE= .200000000E+01 TAIL= .100000000E+01       BW= .233333320E+08
  PTI= .000000000E+01   PR= .155454483E-01      PTO= .148899354E-01
  XIO= .000000000E+01  BTR= .660000000E+02     XLAM= .850000000E+03
 SPEC= .150000000E+02  TVA= .159999996E+00     AVGL= .600000000E+02
* DEFINE AVGL=50.0
* TAM

*TAM*
 SRCE= .200000000E+01 TAIL= .100000000E+01       BW= .233333320E+08
  PTI= .000000000E+01   PR= .454924814E-02      PTO= .840992294E-02
  XIO= .000000000E+01  BTR= .660000000E+02     XLAM= .850000000E+03
 SPEC= .150000000E+02  TVA= .159999996E+00     AVGL= .500000000E+02
* DEFINE AVGL=46.0
* TAM

*TAM*
 SRCE= .200000000E+01 TAIL= .100000000E+01       BW= .233333320E+08
  PTI= .000000000E+01   PR= .864910427E-03      PTO= .670854468E-03
  XIO= .000000000E+01  BTR= .660000000E+02     XLAM= .850000000E+03
 SPEC= .150000000E+02  TVA= .159999996E+00     AVGL= .460000000E+02
```

```
*LIST TAMFILE TEST
  SRCE= .20000E+01         TAIL= .10000E+01         XLEN= .00000E+01
   PTI= .00000E+01         ANGL= .90000E+02           DS= .75000E+02
   XIO= .20000E+01          BTR= .66000E+02           RT= .15000E-07
    BW= .00000E+01         SPEC= .15000E+02         XLAM= .85000E+03
   TVA= .16000E+00          TFP= .10000E+01          TDF= .85000E+02
  *END FILE*
*LIST RAMFILF TEST
  REC TYPE= .30000E+01   GAIN= .10000E+01         XAD = .30000E+00
       XIL = .10000E+03  XIB = .30000E-01         XIS0= .54170E+03
       RES = .55000E+00                           XM  = .00000E+01
       SRMS= .00000E+01  RVB = .00000E+01         XMSC= .00000E+01
       XMA = .00000E+01  BETA= .00000E+01         R   = .10000E+08

  *END FILE*
*LIST FAMFILE TEST
  CORE= .14612E+01         CLAD= .14525E+01          XNA= .16000E+00
    DF= .35000E+02           FP= .35000E+00          DEL= .59500E-02
    AF= .10000E+02         ALFA= .25000E+02         XLMC= .20000E+01
   PNI= .00000E+01          PDD= .00000E+01         PSEC= .00000E+01
    AS= .10000E+01           AC= .20000E+01          XAI= .00000E+01
  *END FILE*
*LIST SAMFILE TEST
    CN= .20000E+01           SN= .00000E+01           XL= .10000E+01
   OPT= .00000E+01         TBER= .10000E-09         TSNR= .10000E-0[illegible]
   XAO= .20000E+01
  *END FILE*
*TAM

  *TAM*
   SRCE= .200000000E+01 TAIL= .100000000E+01        BW= .233333320E+08
    PTI= .000000000E+01   PR= .916026533E-02       PTO= .234502746E-03
    XIO= .000000000E+01  BTR= .660000000E+02      XLAM= .850000000E+03
   SPEC= .150000000E+02  TVA= .159999996E+00      ANGL= .900000000E+02
*FAM

  *FAM*
   DTRA= .102775669E+01 DTER= .707843304E+01      DTOT= .407290843E+01
     BL= .324092800E+08  XLC= .399999943E+03       PTO= .234502746E-03
    PFI= .000000000E+01  PFO= .589043975E-05
   XLAM= .850000000E+03 SPEC= .150000000E+02      ALFA= .250000000E+02
    DEL= .594999921E-02  XNA= .159999996E+00      CORE= .146120000E+01
   CLAD= .145249987E+01  XAI= .159175987E+02      LOSS= .160000000E+02
*RAM
```

```
*RAM*
 TYPE= .300000000E+01      R= .100000000E+08      GAIN= .100000000E+01
 BAND= .500000000E+07   PFD= .589043975E-05       BER= .363332475E-07
  XAD= .299999952E+00 XISW= .541699951E+03        XIL= .100000000E+03
  XIB= .299999975E-01   RES= .549999952E+00       RVB= .300000000E+01
*SAM

******

 AVAILABLE POWER (SOURCE) = .58904E-02

      INPUT COUPLING LOSS = .15918E+02
               FIBER LOSS = .10000E+02
              SPLICE LOSS = .30000E+01
           CONNECTOR LOSS = .40000E+01
     OUTPUT COUPLING LOSS = .20000E+01

   RECEIVER POWER (INPUT) = .58904E-02

                      BER =      .36333E-07

              DESIRED BER =      .10000E-09

         SYSTEM BANDWIDTH = .48343E+07

******
```

```
*DEFINE PFO=0.000005
*RAM

 *RAM*
  TYPE= .800000000E+01      R= .100000000E+08     GAIN= .100000000E+01
  BAND= .500000000E+07  PFO= .499999987E-05      BER= .243285604E-05
   XAD= .299999952E+00 XISQ= .541699951E+03      XIL= .100000000E+03
   XIB= .299999975E-01  RES= .549999952E+00      RVB= .000000000E+01
*DEFINE PFO=0.000004,0.000007,0.000002
*RAM

 *RAM*
  TYPE= .800000000E+01      R= .100000000E+08     GAIN= .100000000E+01
  BAND= .500000000E+07  PFO= .399999954E-05      BER= .128003332E-03
   XAD= .299999952E+00 XISQ= .541699951E+03      XIL= .100000000E+03
   XIB= .299999975E-01  RES= .549999952E+00      RVB= .000000000E+01

 *RAM*
  TYPE= .800000000E+01      R= .100000000E+08     GAIN= .100000000E+01
  BAND= .500000000E+07  PFO= .599999930E-05      BER= .207205950E-07
   XAD= .299999952E+00 XISQ= .541699951E+03      XIL= .100000000E+03
   XIB= .299999975E-01  RES= .549999952E+00      RVB= .000000000E+01
*DEFINE PFO=0.000004,0.000008,0.000002
*ORDER RAM
*SYS

 *RAM*
  TYPE= .800000000E+01      R= .100000000E+08     GAIN= .100000000E+01
  BAND= .500000000E+07  PFO= .399999954E-05      BER= .128003332E-03
   XAD= .299999952E+00 XISQ= .541699951E+03      XIL= .100000000E+03
   XIB= .299999975E-01  RES= .549999952E+00      RVB= .000000000E+01

 *RAM*
  TYPE= .800000000E+01      R= .100000000E+08     GAIN= .100000000E+01
  BAND= .500000000E+07  PFO= .599999930E-05      BER= .207205950E-07
   XAD= .299999952E+00 XISQ= .541699951E+03      XIL= .100000000E+03
   XIB= .299999975E-01  RES= .549999952E+00      RVB= .000000000E+01

 *RAM*
  TYPE= .800000000E+01      R= .100000000E+08     GAIN= .100000000E+01
  BAND= .500000000E+07  PFO= .799999907E-05      BER= .130833211E-12
   XAD= .299999952E+00 XISQ= .541699951E+03      XIL= .100000000E+03
   XIB= .299999975E-01  RES= .549999952E+00      RVB= .000000000E+01
```

```
*GET TAMFILE TEST
 TAMFILE   TEST   RETRIEVED
*GET FAMFILE TEST
 FAMFILE   TEST   RETRIEVED
*GET RAMFILE TEST
 RAMFILE   TEST   RETRIEVED
*GET SAMFILE TEST
 SAMFILE   TEST   RETRIEVED
```

```
* LIST TAMFILE
  SRCE= .20000E+01        TAIL= .10000E+01        XLEV= .00000E+01
   PTI= .00000E+01        ANGL= .90000E+02          DS= .75000E+02
   XIO= .00000E+01         BTR= .66000E+02          RT= .15000E-07
    BW= .00000E+01        SPEC= .15000E+02        XLAM= .85000E+03
   TNA= .16000E+00         TFP= .10000E+01         TDF= .85000E+02
  *END FILE*
* LIST FAMFILE
  CORE= .14612E+01        CLAD= .14525E+01         XNA= .16000E+00
    DF= .85000E+02          FP= .35000E+00         DEL= .59500E-02
    AF= .10000E+02        ALFA= .25000E+02        XLMC= .20000E+01
   PN1= .00000E+01         PDD= .00000E+01        PSEC= .00000E+01
    AS= .10000E+01          AC= .20000E+01         XAI= .00000E+01
  *END FILE*
* LIST RAMFILE
  REC TYPE= .80000E+01    GAIN= .10000E+01        XAD = .30000E+00
      XIL = .10000E+03    XIB = .30000E-01        XISQ= .54170E+03
      RES = .55000E+00                            XM  = .00000E+01
      SRMS= .00000E+01    RNB = .00000E+01        XMSC= .00000E+01
      XMA = .00000E+01    BETA= .00000E+01        R   = .10000E+08

  *END FILE*
* LIST SAMFILE
    CN= .20000E+01          SN= .00000E+01          XL= .10000E+01
   OPT= .00000E+01        TBER= .10000E-09        TSNR= .10000E-09
   XAO= .20000E+01
  *END FILE*
* SYS

  *TAM*
   SRCE= .200000000E+01 TAIL= .100000000E+01        BW= .233333320E+08
    PTI= .000000000E+01   PR= .916026905E-02       PTO= .234502862E-03
    XIO= .000000000E+01  BTR= .660000000E+02      XLAM= .850000000E+03
   SPEC= .150000000E+02  TNA= .159999996E+00      ANGL= .900000000E+02

  *FAM*
   DTRA= .102775598E+01 DTER= .707843208E+01      DTOT= .407290745E+01
     BL= .324092800E+05  XLC= .309999943E+00       PTO= .234502862E-03
    PFI= .000000000E+01  PFO= .589044521E-05
   XLAM= .850000000E+03 SPEC= .150000000E+02      ALFA= .247879939E-09
    DEL= .594999981E-02  XNA= .159999996E+00      CORE= .146120000E+01
   CLAD= .145249987E+01  XAI= .159176006E+02      LOSS= .160000000E+02

  *RAM*
   TYPE= .800000000E+01    R= .100000000E+08      GAIN= .100000000E+01
   BAND= .500000000E+07  PFO= .589044521E-05       BER= .363320680E-07
    XAD= .299999952E+00 XISQ= .541699951E+03       XIL= .100000000E+03
    XIB= .299999975E-01  RES= .549999952E+00       RNB= .000000000E+01
```

```
******

AVAILABLE POWER (SOURCE) = .58934E-05

    INPUT COUPLING LOSS = .15918E+02
             FIBER LOSS = .10000E+02
            SPLICE LOSS = .00000E+01
         CONNECTOR LOSS = .40000E+01
   OUTPUT COUPLING LOSS = .20000E+01

 RECEIVER POWER (INPUT) = .58934E-05

                    BER =      .36332E-07

            DESIRED BER =      .10000E-09

       SYSTEM BANDWIDTH = .48343E+07

******

*DEFINE GAIN=1.0,10.0,9.0
*DEFINE PFO=.4E-5,0.5E-5,0.1E-5
*RAM

*RAM*
  TYPE= .800000000E+01      R= .100000000E+08     GAIN= .100000000E+01
  BAND= .500000000E+07    PFO= .399999954E-05      BER= .128003012E-03
   XAD= .299999952E+00   XISQ= .541699951E+03      XIL= .100000000E+03
   XIB= .299999975E-01    RES= .549999952E+00      RNB= .000000000E+01

*RAM*
  TYPE= .800000000E+01      R= .100000000E+08     GAIN= .100000000E+01
  BAND= .500000000E+07    PFO= .499999987E-05      BER= .243285649E-05
   XAD= .299999952E+00   XISQ= .541699951E+03      XIL= .100000000E+03
   XIB= .299999975E-01    RES= .549999952E+00      RNB= .000000000E+01

*RAM*
  TYPE= .800000000E+01      R= .100000000E+08     GAIN= .100000000E+02
  BAND= .500000000E+07    PFO= .399999954E-05      BER= .000000000E+01
   XAD= .299999952E+00   XISQ= .541699951E+03      XIL= .100000000E+03
   XIB= .299999975E-01    RES= .549999952E+00      RNB= .000000000E+01

*RAM*
  TYPE= .800000000E+01      R= .100000000E+08     GAIN= .100000000E+02
  BAND= .500000000E+07    PFO= .499999987E-05      BER= .000000000E+01
   XAD= .299999952E+00   XISQ= .541699951E+03      XIL= .100000000E+03
   XIB= .299999975E-01    RES= .549999952E+00      RNB= .000000000E+01
*DEFINE GAIN=0.0,10.0,5.0
*RAM
```

```
*RAM*
 TYPE= .800000000E+01      R= .100000000E+08    GAIV= .000000000E+01
 BAND= .500000000E+07   PFO= .399999954E-05      BER= .128003312E-03
  XAD= .299999952E+00  XISQ= .541699951E+03      XIL= .100000000E+03
  XIB= .299999975E-01   RES= .549999952E+00      RVB= .000000000E+01

*RAM*
 TYPE= .800000000E+01      R= .100000000E+08    GAIV= .000000000E+01
 BAND= .500000000E+07   PFO= .499999987E-05      BER= .243285649E-05
  XAD= .299999952E+00  XISQ= .541699951E+03      XIL= .100000000E+03
  XIB= .299999975E-01   RES= .549999952E+00      RVB= .000000000E+01

*RAM*
 TYPE= .800000000E+01      R= .100000000E+08    GAIV= .500000000E+01
 BAND= .500000000E+07   PFO= .399999954E-05      BER= .000000000E+01
  XAD= .299999952E+00  XISQ= .541699951E+03      XIL= .100000000E+03
  XIB= .299999975E-01   RES= .549999952E+00      RVB= .000000000E+01

*RAM*
 TYPE= .800000000E+01      R= .100000000E+08    GAIV= .500000000E+01
 BAND= .500000000E+07   PFO= .499999987E-05      BER= .000000000E+01
  XAD= .299999952E+00  XISQ= .541699951E+03      XIL= .100000000E+03
  XIB= .299999975E-01   RES= .549999952E+00      RVB= .000000000E+01

*RAM*
 TYPE= .800000000E+01      R= .100000000E+08    GAIV= .100000000E+02
 BAND= .500000000E+07   PFO= .399999954E-05      BER= .000000000E+01
  XAD= .299999952E+00  XISQ= .541699951E+03      XIL= .100000000E+03
  XIB= .299999975E-01   RES= .549999952E+00      RVB= .000000000E+01

*RAM*
 TYPE= .800000000E+01      R= .100000000E+08    GAIV= .100000000E+02
 BAND= .500000000E+07   PFO= .499999987E-05      BER= .000000000E+01
  XAD= .299999952E+00  XISQ= .541699951E+03      XIL= .100000000E+03
  XIB= .299999975E-01   RES= .549999952E+00      RVB= .000000000E+01
*CLEAR
```

```
*GET TAMFILE TEST
 TAMFILE    TEST   RETRIEVED
*GET FAMFILE TEST
 FAMFILE    TEST   RETRIEVED
*GET RAMFILE TEST
 RAMFILE    TEST   RETRIEVED
*GET SAMFILE TEST
 SAMFILE    TEST   RETRIEVED
*DEFINE TVA=.10,.30,0.1
*TAM

 *TAM*
  SRCE= .200000000E+01 TAIL= .100000000E+01        BW= .233333320E+08
   PTI= .000000000E+01   PR= .916026905E-02      PTO= .916026620E-04
   XIO= .000000000E+01  BTR= .660000000E+02     XLAM= .850000000E+03
  SPEC= .150000000E+02  TVA= .999999940E-01     AVGL= .900000000E+02

 *TAM*
  SRCE= .200000000E+01 TAIL= .100000000E+01        BW= .233333320E+08
   PTI= .000000000E+01   PR= .916026905E-02      PTO= .366410648E-03
   XIO= .000000000E+01  BTR= .660000000E+02     XLAM= .850000000E+03
  SPEC= .150000000E+02  TVA= .199999985E+00     AVGL= .900000000E+02

 *TAM*
  SRCE= .200000000E+01 TAIL= .100000000E+01        BW= .233333320E+08
   PTI= .000000000E+01   PR= .916026905E-02      PTO= .824423740E-03
   XIO= .000000000E+01  BTR= .660000000E+02     XLAM= .850000000E+03
  SPEC= .150000000E+02  TVA= .299999952E+00     AVGL= .900000000E+02
*END
 STOP
```

SECTION 4.0

DESIGN CURVE EXAMPLES

4.0 DESIGN CURVE EXAMPLES

4.1 Introduction

One of the useful outputs of FODAP consists of the design curves that can be generated by incrementing various input data. Unfortunately, because of the lack of component standardization in the fiber optic field, one cannot construct design curves that cover all situations. Once components are standardized, FODAP can then be used to generate generalized curves. It would simply be a matter of using input files consisting of data for the standardized components. In the mean time, one can construct curves for specific design situations. Examples of such curves are given in the sections that follow. Sample curves appropriate to transmitter, fiber or receiver design are given in Sections 4.2, 4.3 and 4.4, respectively. Section 4.5 gives examples of systems-level design curves.

4.2 Transmitter Design Curves

One important operation in TAM calculates the input coupling loss from the source to a fiber pigtail. The input coupling loss is a function of both source and fiber parameters. It is interesting and informative to see how this loss is affected as the pertinent parameters are varied. Such results can help the designer to select the proper device to maximize source-to-fiber coupling. The largest portion of the input coupling loss is usually L_{NA} as defined in equation (2.4.1.1-1). Figure 4.2-1 shows how L_{NA} varies as a function of θ_n, the null angle. A single fiber with an NA of 0.18 was assumed. The value of L_{NA} at $\theta_n = 90^\circ$ was calculated using equation (2.4.1.1-4). This curve shows that as the source output beam becomes more sharply focused, the loss is reduced. An important point should be made here: One does not necessarily reduce the total loss simply by focusing the beam. The emission angle reduction caused by focusing is accompanied by a source area magnification. Once the magnified area becomes larger than the fiber core area, the unintercepted illumination loss (equation 2.4.1.1-2) becomes important. Thus, one wishes to reduce the emission angle just to the point where $A_s = A_F$. If $A_s \geq A_F$ to begin with, then focusing has no effect.

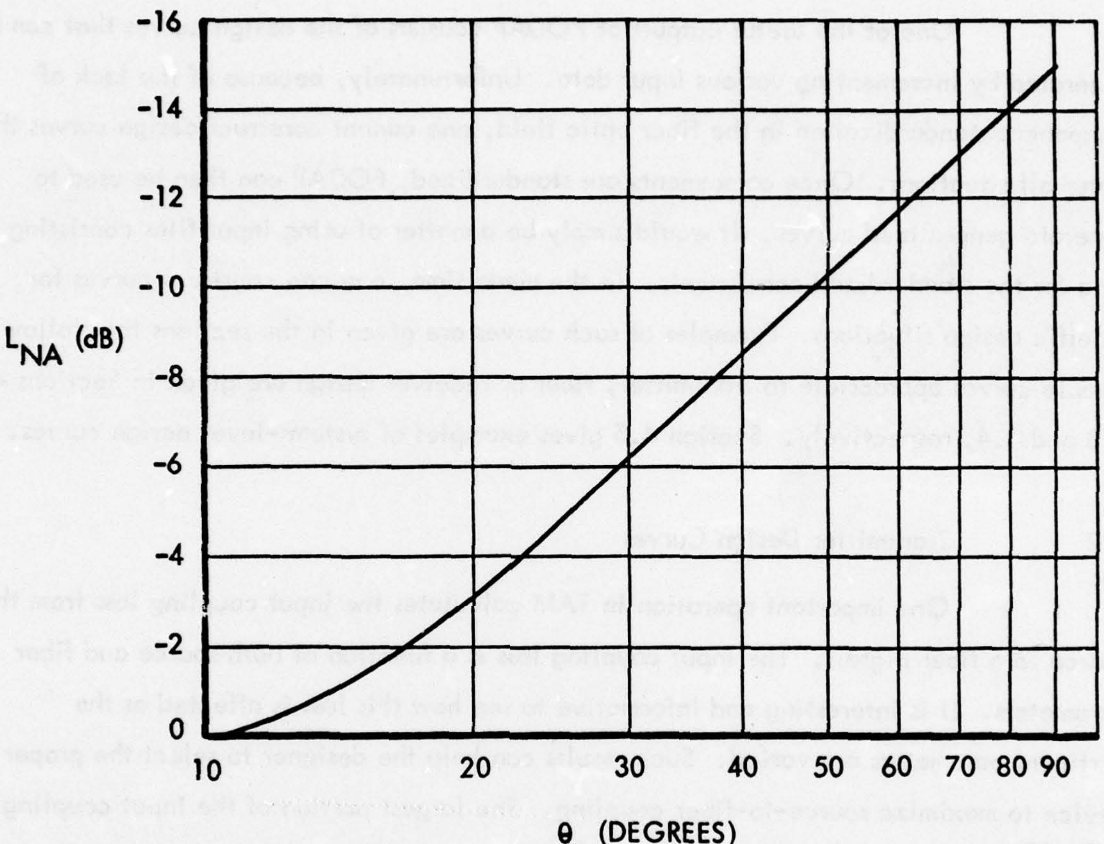


Figure 4.2-1. L_{NA} Versus θ_n

While the above mentioned figure illustrates how source parameters affect the coupling loss, fiber parameters can also have a large effect. In particular, the fiber numerical aperature has a large influence on coupling loss. Figure 4.2-2 shows a plot of L_{NA} as a function of NA for the case of a Lambertian emitter. For purposes of comparison, the loss associated with numerical aperature effects for an edge emitting LED is also plotted. Note the superior performance of the edge-type emitter. There is roughly 7 dB less loss. This is the result of the fact that the edge emitter has an elliptical emission pattern. That is, it is a "compressed" Lambertian pattern in one plane and a normal Lambertian pattern in the other plane. The slight focusing gives a lower loss.

4.3 Fiber Design Curves

One of the more useful fiber design curves is a plot of the rms dispersion per unit length $\hat{\sigma}$ (assuming no mode coupling effects) as a function of the fiber profile parameter α. Figure 4.3-1 shows this curve. It was generated using the equations for inter- and intramodal dispersion given in Section 2.3. Two specific cases are considered; one, a 15 nm spectral width LED and the other a 1 nm spectral width ILD. Both were assumed to emit at a peak wavelength of 850 nm. The fiber parameters, as shown on the figure were those appropriate to a 0.16 NA fiber. Note how at the optimum α, the total dispersion is minimized for both cases. However, because of a lower intramodal dispersion for the ILD (due to its narrower spectral width), the total dispersion resulting from its use is considerably less than for the LED. The optimum α shown is the design goal for manufacturers of graded index fibers. However, since the fiber making art is relatively new, these manufacturers are only presently able to make fibers with profiles of 10% of optimum.

When mode coupling effects are accounted for, the situation gets more interesting. The designer is able to examine trade-off's between excess loss and increased bandwidth. Curves showing total dispersion σ_{total} as a function of fiber length are shown in Figures 4.3-2 through 4.3-5. The parameters used to generate Figure 4.3-1 are used in these figures with the addition that excess losses associated with mode coupling of 0, 2, 6 and 10 dB/km are used.

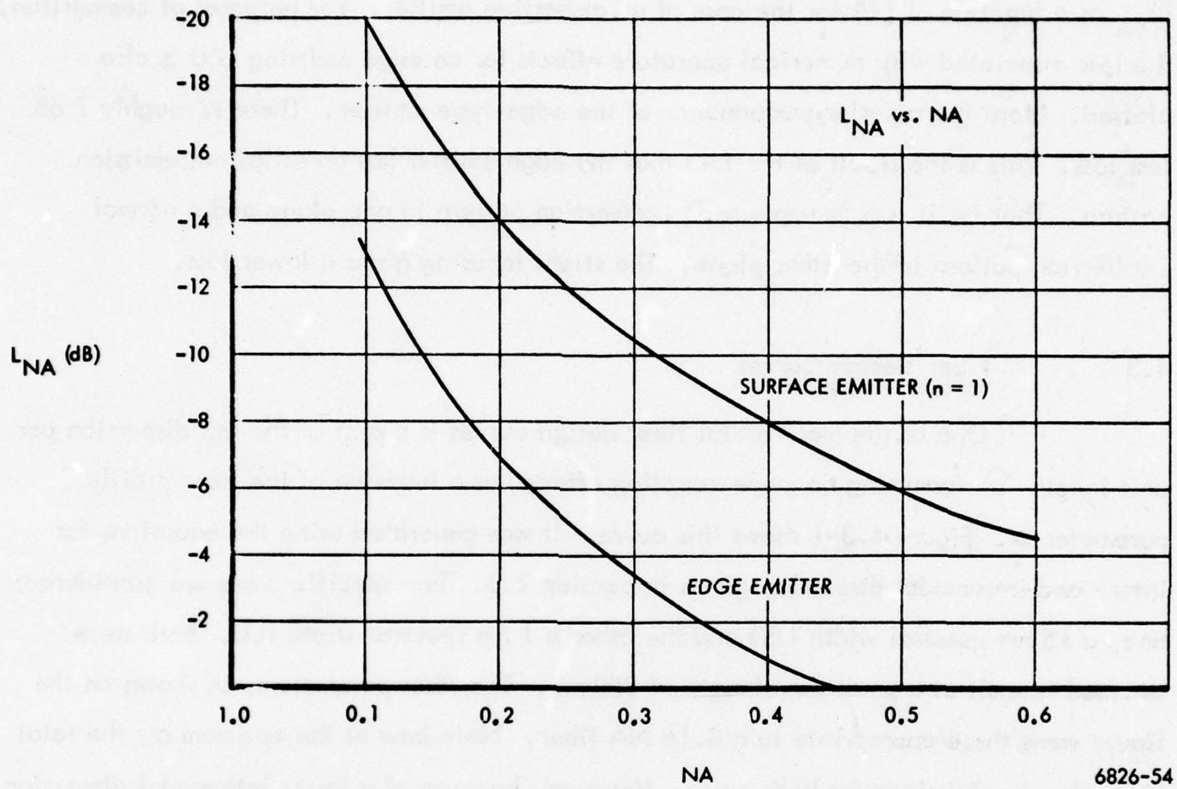


Figure 4.2-2. L_{NA} Versus NA

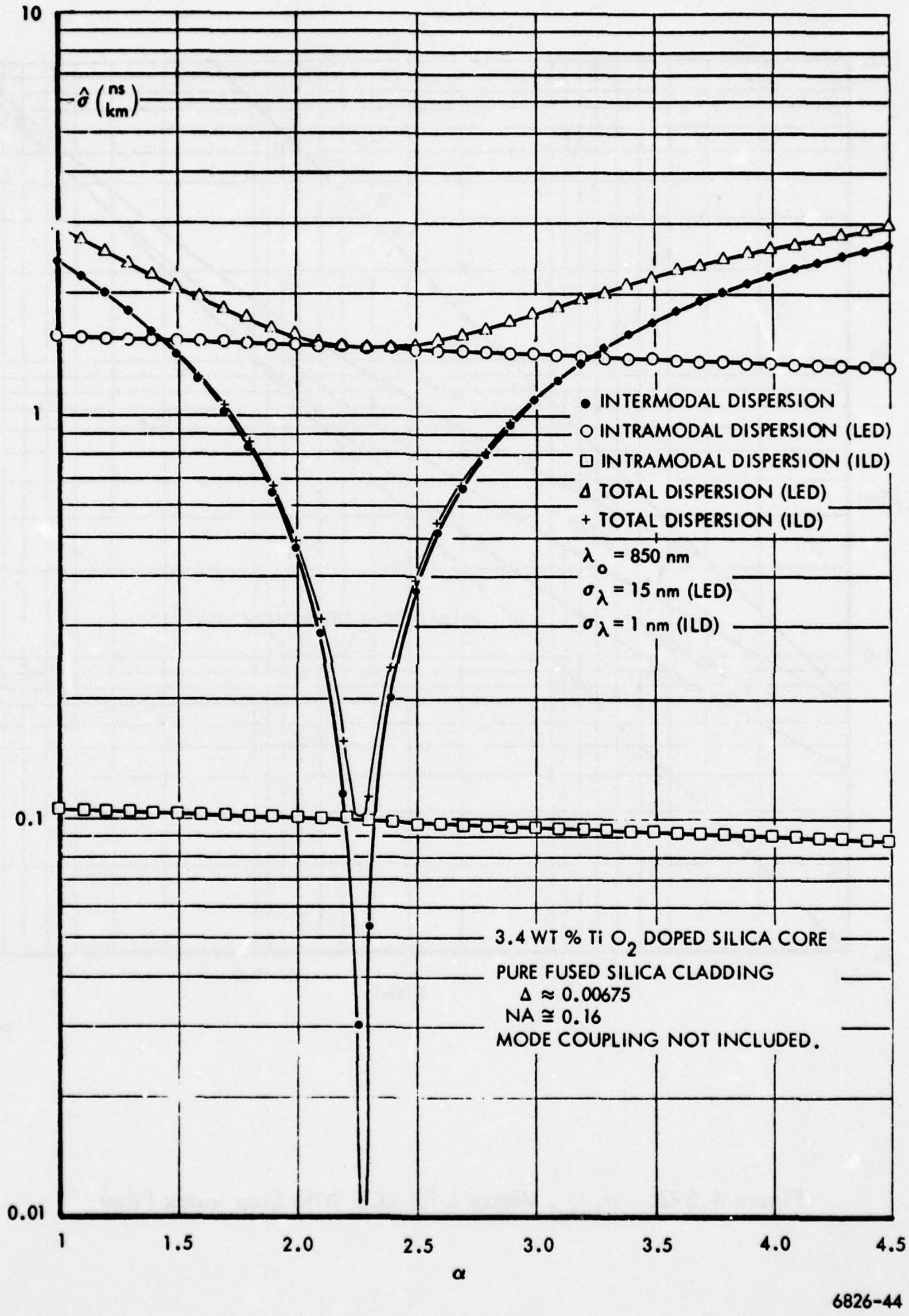


Figure 4.3-1. $\hat{\sigma}$ Versus α

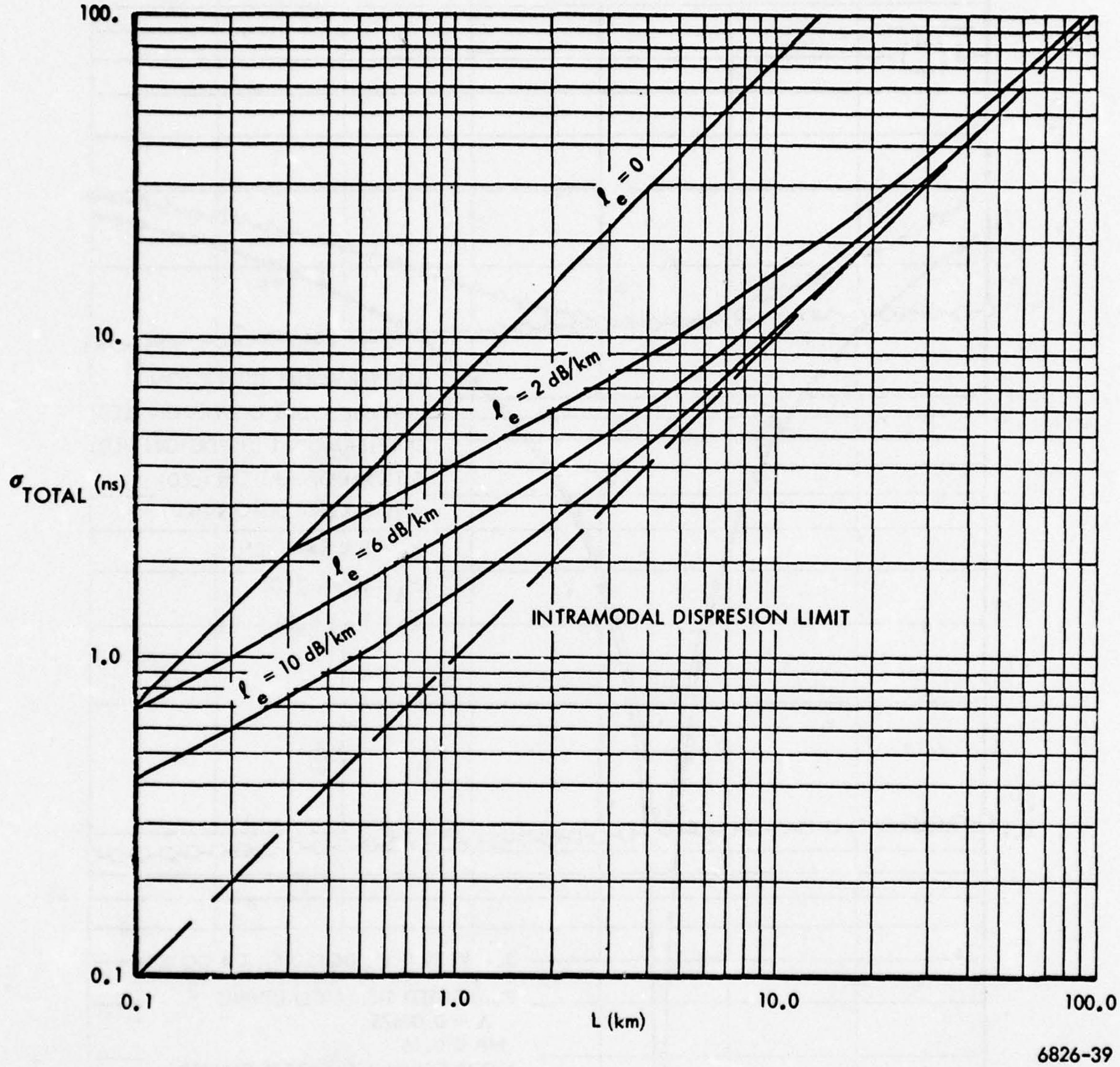


Figure 4.3-2. σ_{total} Versus L for LED With Step Index Fiber

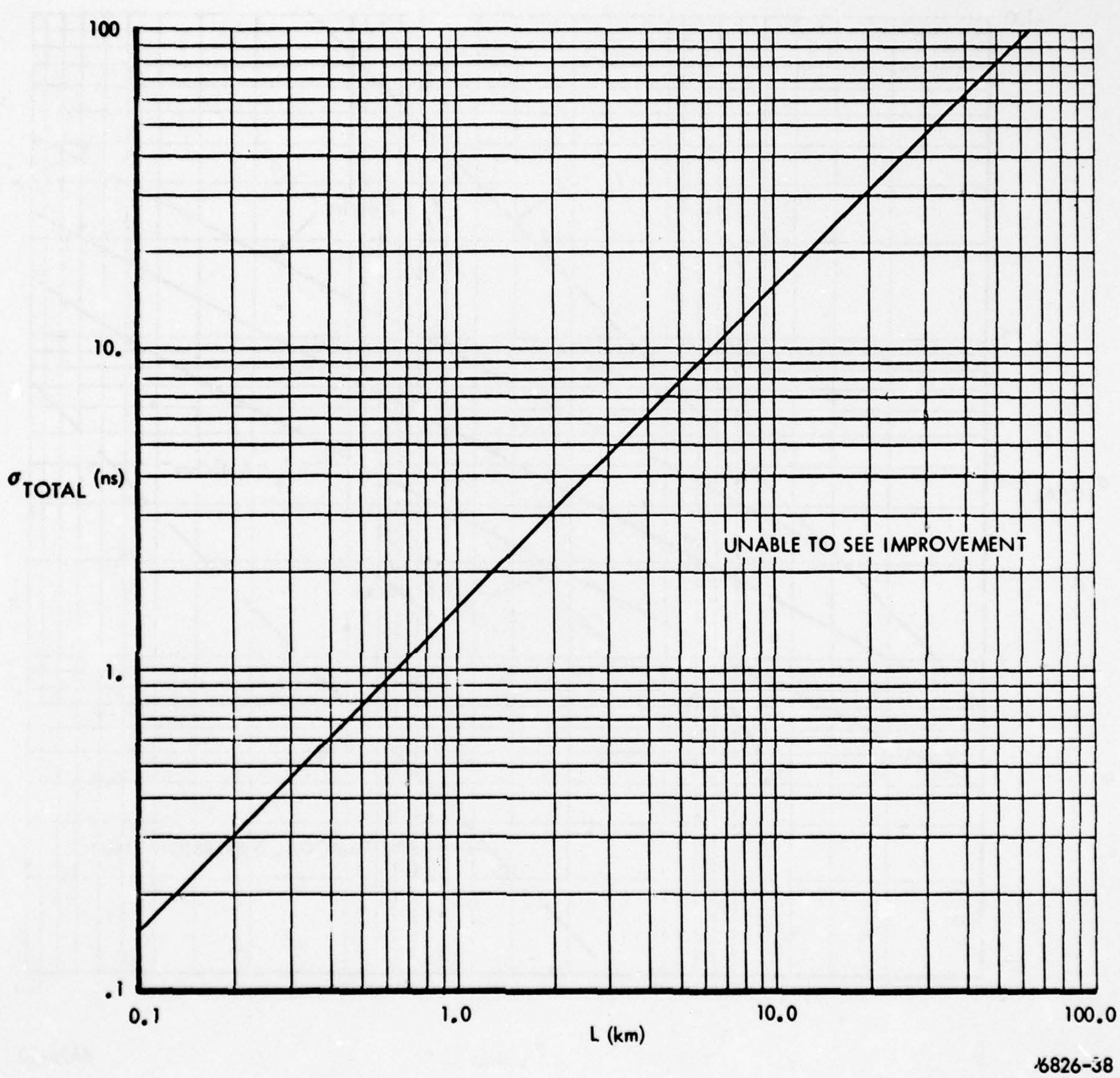
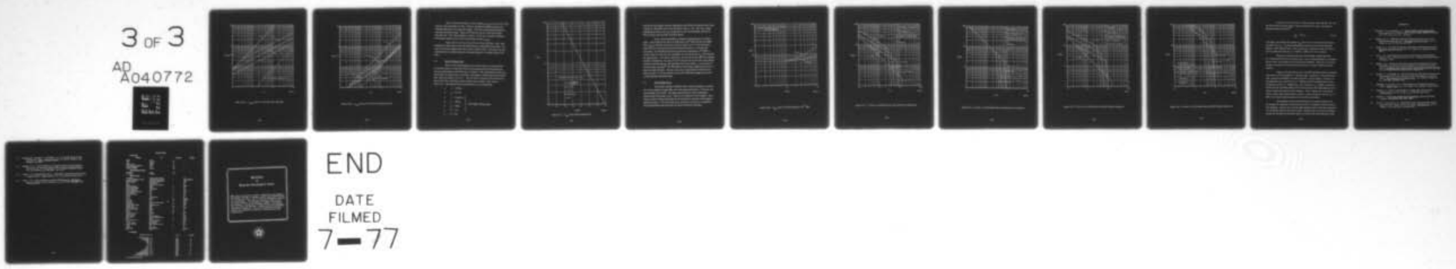


Figure 4.3-3. σ_{total} Versus L for LED With Graded Index Fiber

AD-A040 772 HARRIS CORP MELBOURNE FLA ELECTRONIC SYSTEMS DIV F/6 20/6
FIBER OPTICS DESIGN AID PACKAGE.(U)
MAY 77 I B SLAYTON, D J ESTAPA, J R JONES F30602-76-C-0246
UNCLASSIFIED RADC-TR-77-163 NL

3 OF 3
AD A040772



END
DATE FILMED
7-77

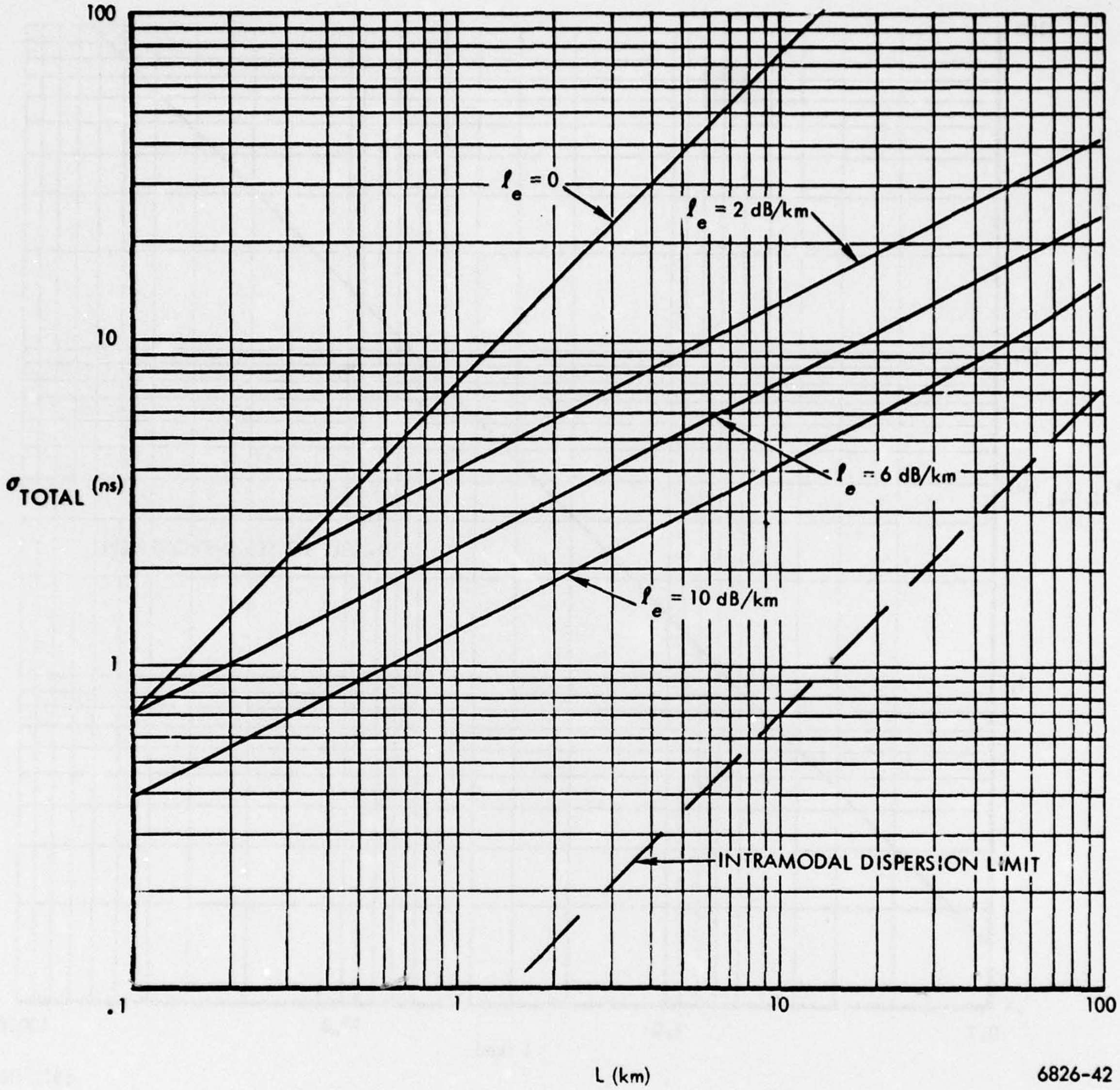


Figure 4.3-4. σ_{total} Versus L for ILD With Step Index Fiber

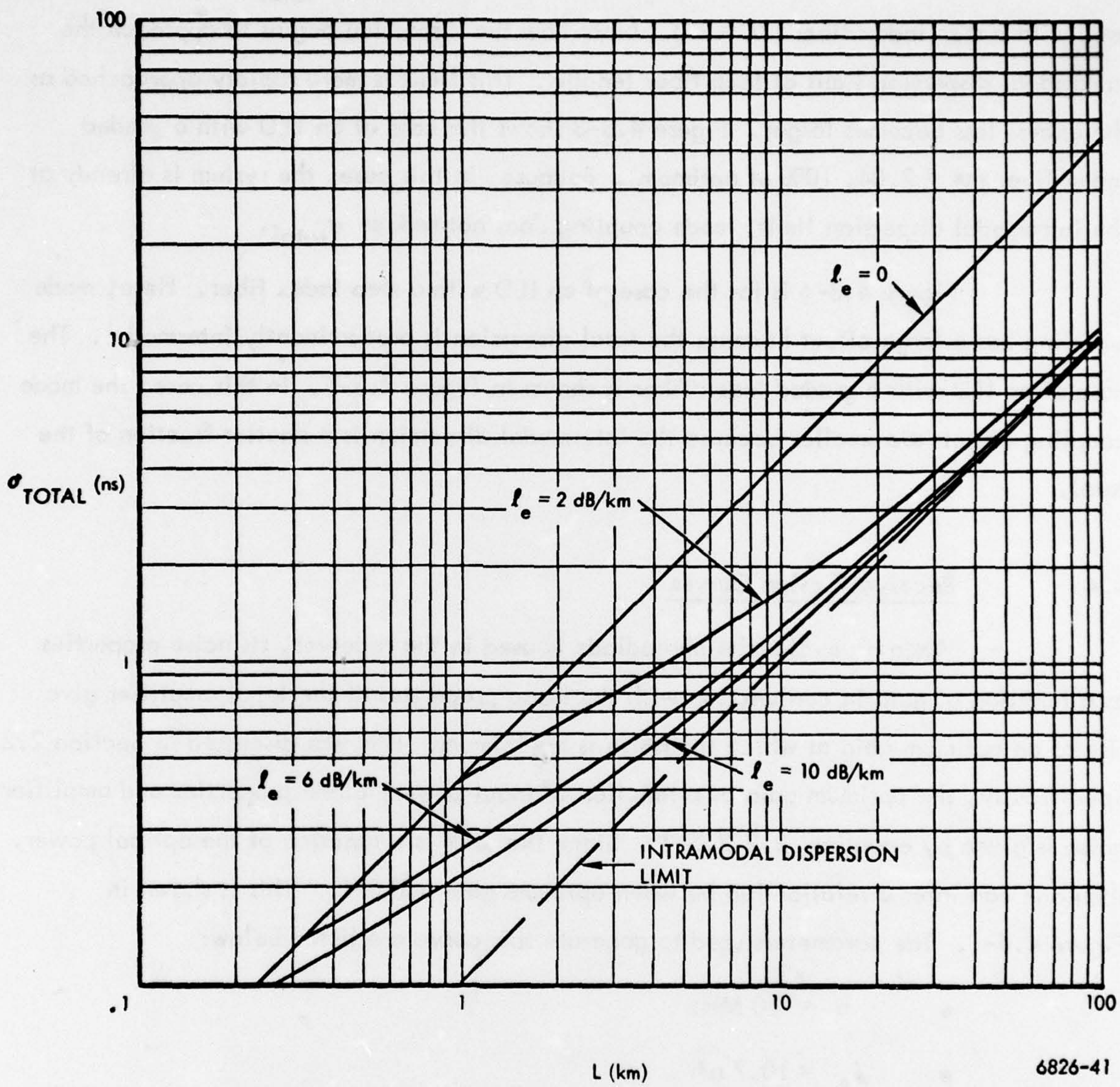


Figure 4.3-5. σ_{total} Versus L for ILD With Graded Index Fiber

Figure 4.3-2 shows the effect of mode coupling on σ_{total} for the case of an LED used with a step index fiber (α = 25). Note how the dispersion begins to approach the intramodal dispersion limit at long fiber lengths. This limit is more rapidly approached as the excess loss becomes larger. Figure 4.3-3 shows the case of an LED with a graded index fiber (α = 2.04, 10% of optimum). Because, in this case, the system is already at the intramodal dispersion limit, mode coupling does not reduce σ_{total}.

Figure 4.3-4 is for the case of an ILD with a step index fiber. Here, mode coupling has a large effect because the total dispersion is predominantly intermodal. The case of an ILD with a graded index fiber is shown in Figure 4.3-5. In this case, the mode coupling effects are smaller because the intermodal dispersion is a smaller fraction of the total.

4.4 Receiver Design Curves

When an avalanche photodiode is used in the receiver, its noise properties as a function of gain in conjunction with the noise properties of the input amplifier give rise to an optimum gain at which the noise is a minimum. This was discussed in Section 2.2. Specifically, the optimum gain as a function of input power, diode properties and amplifier noise is given by equation 2.2.2.5-1. Since the SNR is a function of the optical power, then one can infer a relationship between optimum gain and SNR. This is shown in Figure 4.4-1. The parameters used to generate this curve are listed below:

- b = 10 MHz
- i_A = 10.7 nA
- r = 0.65 A/W
- I_L = 100 nA
- I_b = 50 pA
- D = 0.3
- M = 0.8

RCA C30817 APD parameters (r, I_L, I_b, D)

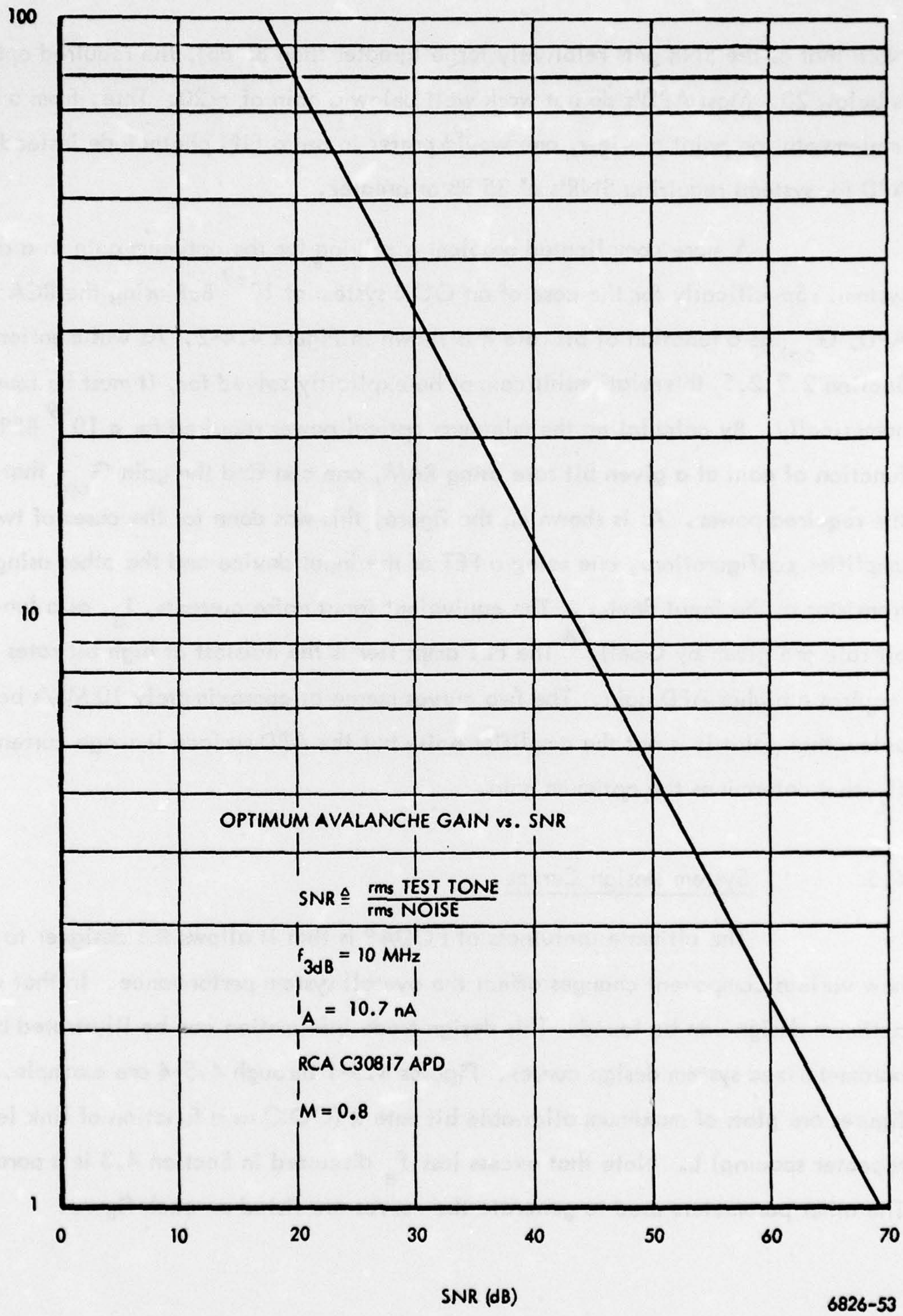


Figure 4.4-1. G_{opt} Versus SNR for Baseband IM

Note that as the SNR gets relatively large (greater than 35 dB), the required optimum gain is below 20. Most APD's do not work well below a gain of ~20. Thus, from a design implementation point of view, one would prefer to use a PIN photodiode instead of an APD for systems requiring SNR's of 35 dB or greater.

A more complicated problem is solving for the optimum gain in a digital system. Specifically for the case of an OOK system at 10^{-9} BER using the RCA C30817 APD, G_{opt} as a function of bit rate R is shown in Figure 4.4-2. As was mentioned in Section 2.2.2.5, this relationship cannot be explicitly solved for, it must be found numerically. By calculating the minimum optical power required for a 10^{-9} BER as a function of gain at a given bit rate using RAM, one can find the gain G_{opt} that minimizes the required power. As is shown on the figure, this was done for the cases of two different amplifier configurations, one using a FET as the input device and the other using a bipolar transistor as the input device. The equivalent input noise currents, i_a, as a function of bit rate are given by Goeli.[14] The FET amplifier is the noisiest at high bit rates and so it requires a higher APD gain. The two curves merge at approximately 10 Mb/s because below that point it is not the amplifier noise but the APD surface leakage current noise (I_L) that determines the optimum gain.

4.5 System Design Curves

The ultimate usefulness of FODAP is that it allows the designer to examine how various component changes affect the overall system performance. In that way, the optimum design can be found. This design trade information can be illustrated by parameterized system design curves. Figures 4.5-1 through 4.5-4 are examples. These figures are plots of maximum allowable bit rate R (OOK) as a function of link length (or repeater spacing) L. Note that excess loss ℓ_e discussed in Section 4.3 is a parameter. The other parameters used to generate the curves are listed on each figure.

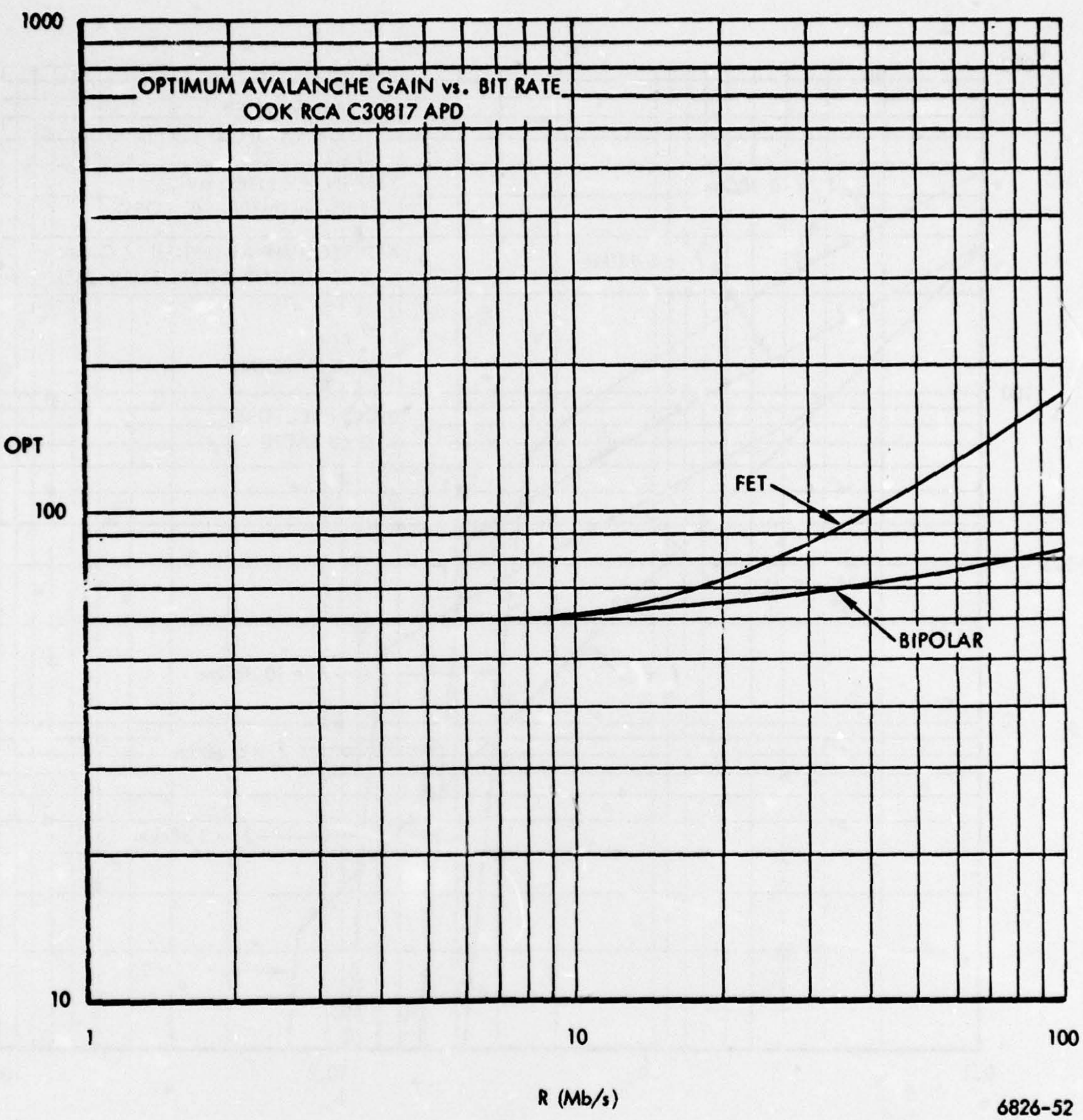


Figure 4.4-2. G_{opt} Versus R for OOK Modulation (10^{-9} BER)

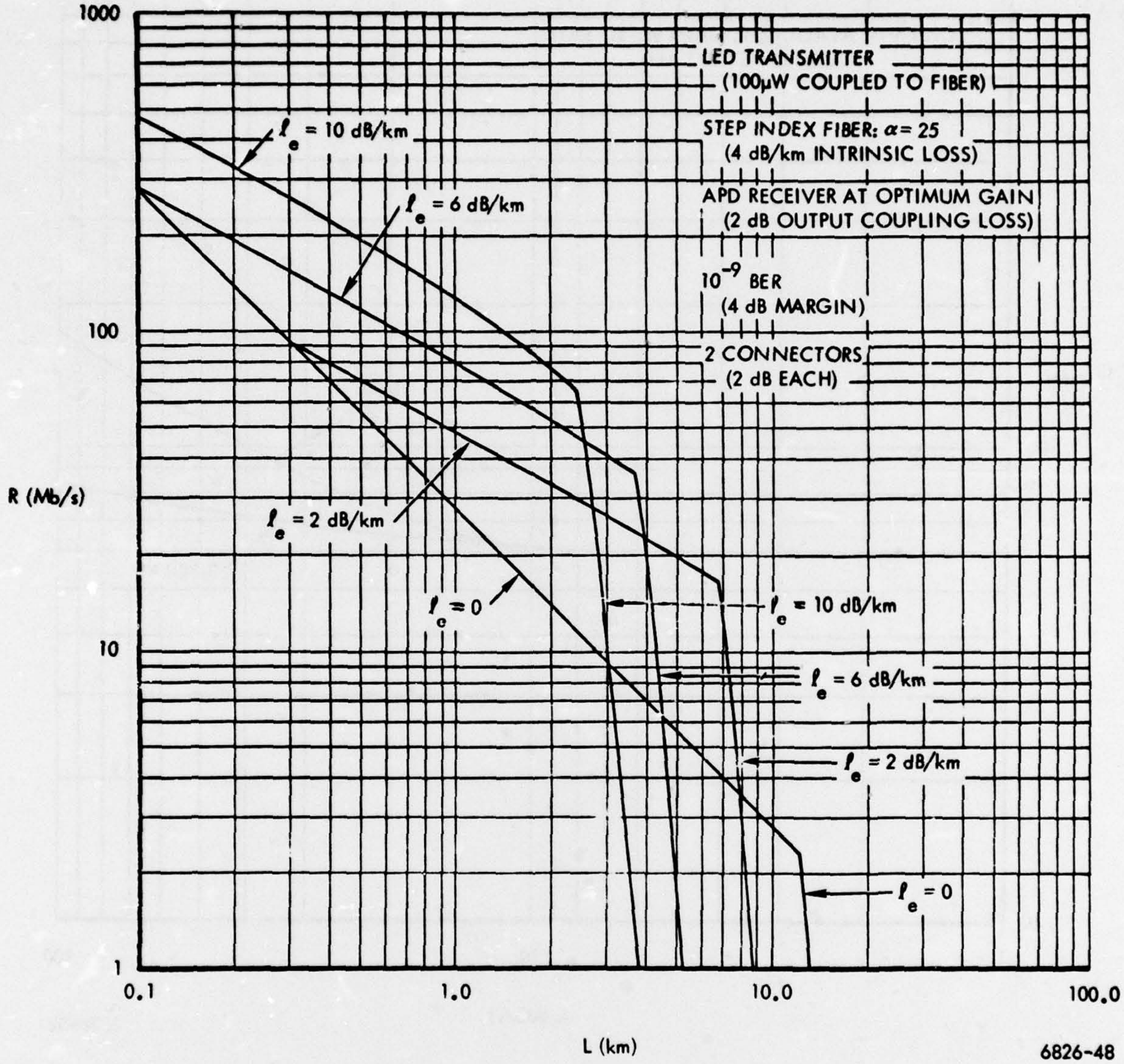


Figure 4.5-1. R Versus L for LED/Step Index Fiber/APD System Combination

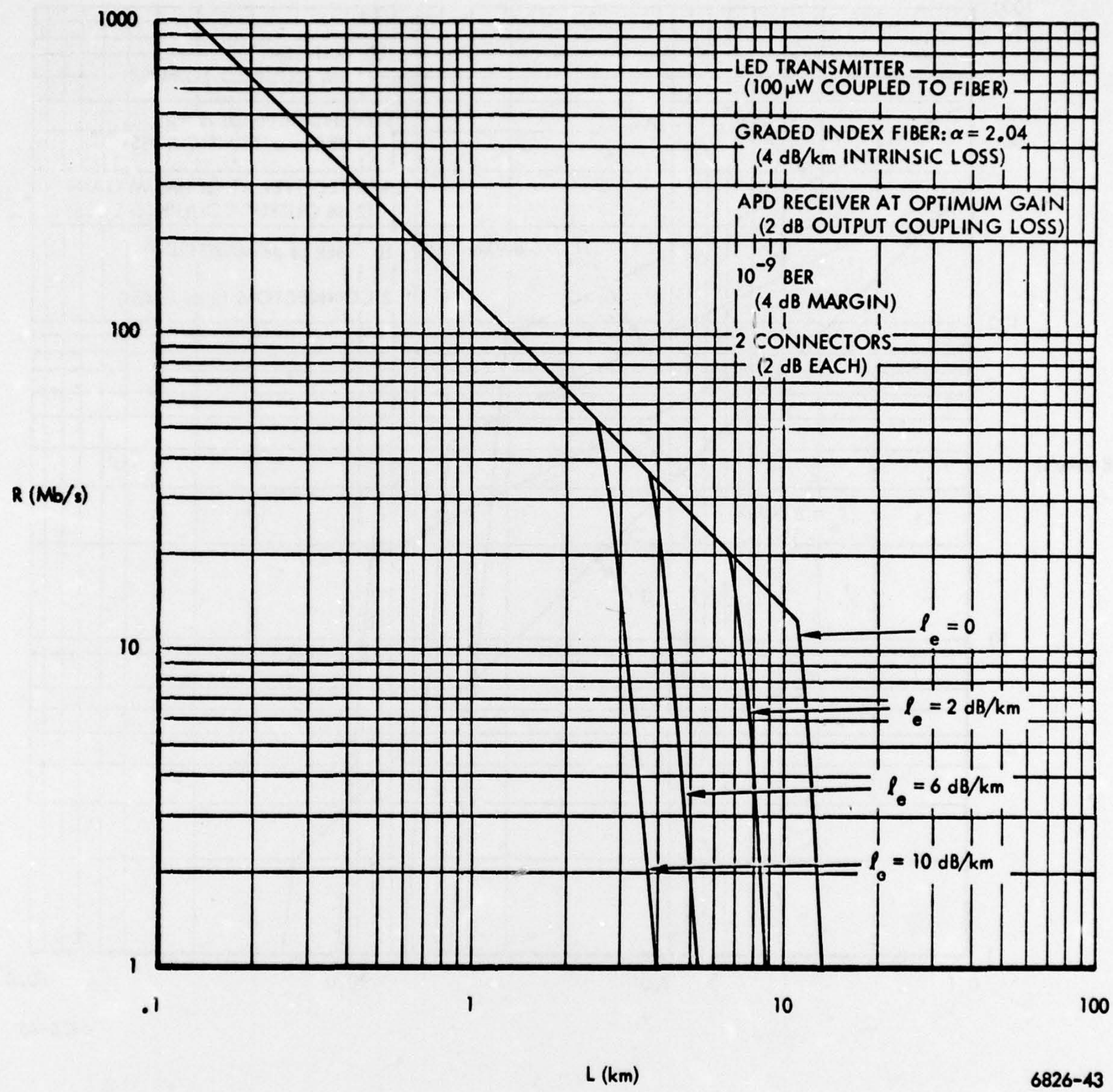


Figure 4.5-2. R Versus L for LED/Graded Index Fiber/APD System Combination

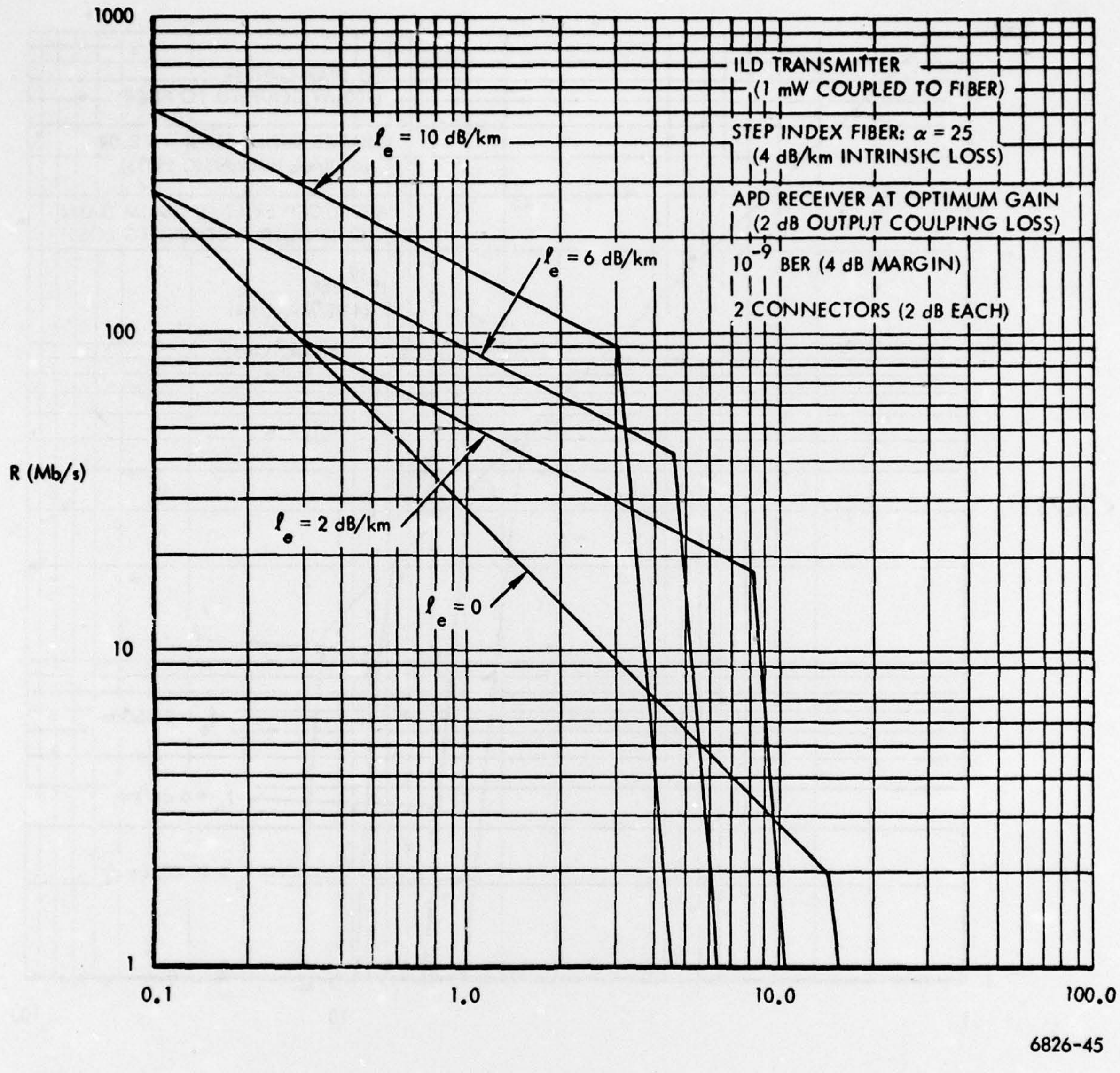


Figure 4.5-3. R Versus L for ILD/Step Index Fiber/APD System Combination

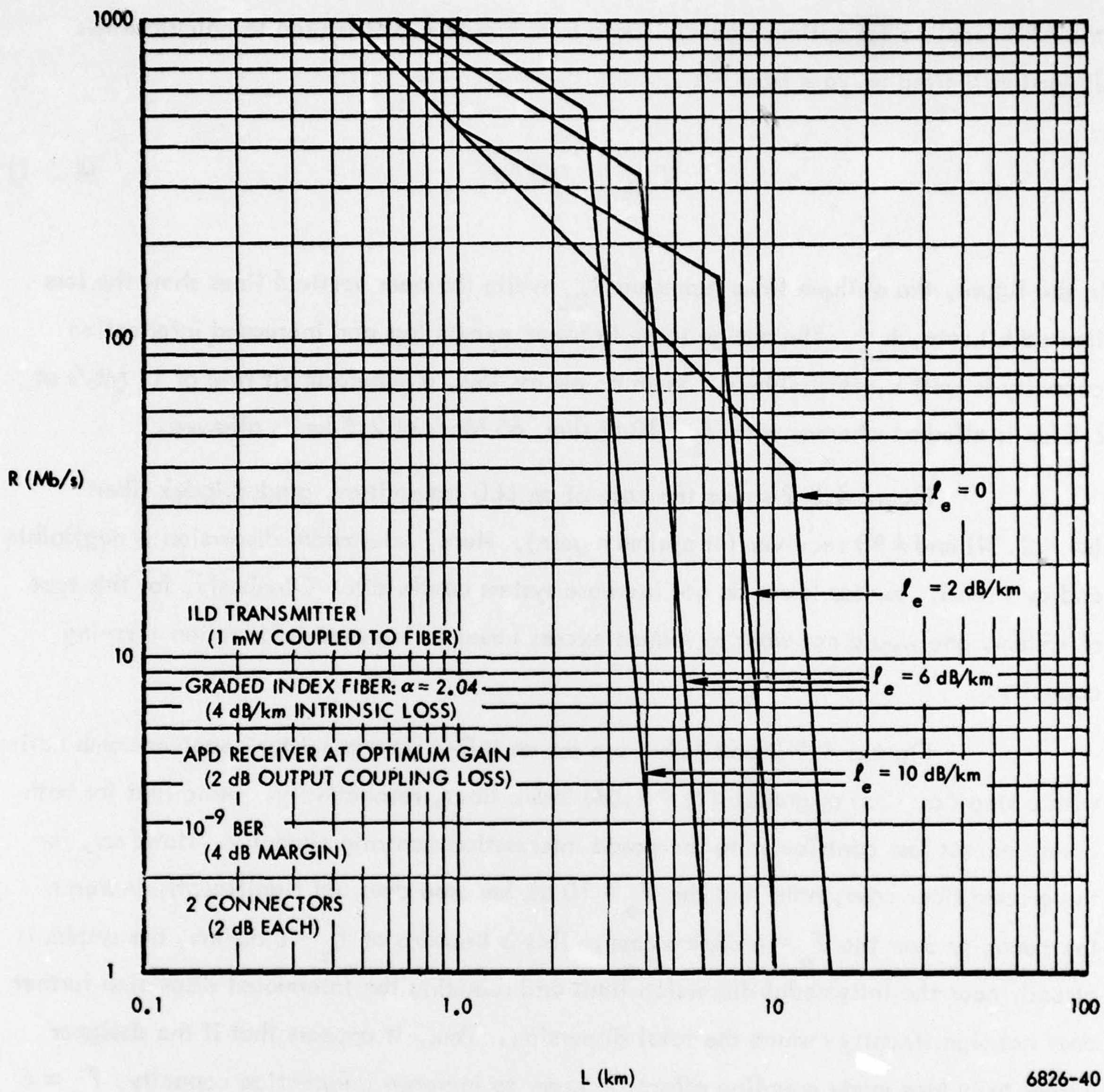


Figure 4.5-4. R Versus L for ILD/Graded Index Fiber/APD System Combination

Figure 4.5-1 is for the case of an LED transmitter, step index fiber (α = 25) and APD receiver (at optimum gain). Data from Figure 4.3-2 is used to calculate the dispersion limited bit rate from[15]

$$R_{DL} = 0.2/\sigma_t \qquad (4.5\text{-}1)$$

In the figure, the oblique lines represent R_{DL} while the near vertical lines show the loss limited bit rate, R_{LL}. The design trade between excess loss and increased information capacity is well illustrated here. With no excess loss, a maximum bit rate of 11 Mb/s at 2.5 km is allowed whereas with ℓ_e = 10 dB/km, 65 Mb/s at 2.5 km is allowed.

Figure 4.5-2 shows the case of an LED transmitter, graded index fiber (α = 2.04) and APD receiver (at optimum gain). Here, intermodal dispersion is negligible and as a result, excess losses do not increase system bandwidth. Obviously, for this type of system, one would not want to induce excess losses to increase information carrying capacity.

Figures 4.5-3 and 4.5-4 are for an ILD/APD transmitter-receiver combination with a step (α = 25) or graded (α = 2.04) index fiber, respectively. Note that for both cases, excess loss contributes to increased information carrying capacity. However, for the graded fiber case, note that the ℓ_e = 10 dB/km case does not significantly increase the capacity over the ℓ_e = 6 dB/km case. This is because at ℓ_e = 6 dB/km, the system is already near the intramodal dispersion limit and reducing the intermodal dispersion further does not significantly reduce the total dispersion. Thus, it appears that if the designer wishes to induce mode coupling effects in order to increase information capacity, $\ell_e \simeq 6$ dB/km is the maximum desired excess loss for this particular situation.

Unfortunately, the above discussed figures are not general design curves because specific components were assumed which may not correspond to those selected by other designers. However, the curves are approximately general in that other components are not radically different from those assumed. Therefore, these curves give the designer insights into the potential information capacity of optical fiber data transmission systems.

REFERENCES

1. McDevitt, F. R., and Slayton, I. B., Optical Cable Communications Study, Harris Corporation Final Report, RADC Contract Number F30602-74-C-0193, July 1975, RADC-TR-75-187, (A016846).

2. Hubbard, W. M., "Utilization of Optical-Frequency Carriers for Low- and Moderate-Bandwidth Channels", Bell System Technical Journal, Vol. 52, Number 5, pp. 731-765, May-June 1973.

3. Holden, W. S., "An Optical-Frequency Pulse-Position Modulation Experiment", Bell System Technical Journal, Vol. 54, Number 2, pp. 285-296, February 1975.

4. Jones, J. R., "An Analog Optical Communications Channel Using Wideband Subcarrier FM", Conference Record, Vol. III., National Telecommunications Conference, 1976.

5. Patisaul, C. R., "Signal-To-Noise Ratio Improvement of Analog Pulse-Position Modulation Over Analog Intensity Modulation in Optical Communication Links", Proceedings of the Society of Photo-Optical Instrumentation Engineers, Vol. 63, Guided Optical Communications, August 1975.

6. Takasaki, Y., and Maeda, M., "Receiver Designs for Fiber Optic Communications Optimization in Terms of Excess Noise Factors That Depend on Avalanche Gains", IEEE Transactions on Communications, Vol. COM-24, Number 12, pp. 1343-1346, December 1976.

7. Olshansky, R. and Kerk, D. B., "Pulse Broadening in Graded-Index Optical Fibers", Applied Optics, Vol. 15, Number 2, pp. 483-491, February 1976.

8. Olshansky, R., "Mode Coupling Effects in Graded-Index Optical Fibers", Applied Optics, Vol. 14, Number 4, pp. 935-945, April 1975.

9. Papoulis, A., The Fourier Integral and Its Applications, McGraw-Hill Book Company, Incorporated, New York, 1962.

10. DiVita, P. and Vannucci, R., "Multimode Optical Waveguides With Graded Refractive Index: Theory of Power Launching", Applied Optics, Vol. 15, Number 11, pp. 2765-2772, November 1976.

11. Ettenberg, M., Kressel, H., and Wittke, J. P., "Very High Radiance Edge-Emitting LED", IEEE J. Quantum Electron, Vol. QE-12, Number 6, pp. 360-364, June 1976.

12. Brackett, C. A., "On the Efficiency of Coupling Light From Stripe Geometry GaAs Lasers into Multimode Optical Fibers", Journal of Applied Physics, Vol. 45, Number 6, pp. 2636-2637, June 1974.

13. Cohen, L. G., and Schneider, M. U., "Microlenses for Coupling Junction Lasers to Optical Fibers", Applied Optics, Vol. 13, pp. 89-94, January 1974.

14. Goell, J. E., "Input Amplifiers for Optical PCM Receivers", Bell System Technical Journal, Vol. 53, Number 9, pp. 1771-1794, November 1974.

METRIC SYSTEM

BASE UNITS:

Quantity	Unit	SI Symbol	Formula
length	metre	m	...
mass	kilogram	kg	...
time	second	s	...
electric current	ampere	A	...
thermodynamic temperature	kelvin	K	...
amount of substance	mole	mol	...
luminous intensity	candela	cd	...

SUPPLEMENTARY UNITS:

plane angle	radian	rad	...
solid angle	steradian	sr	...

DERIVED UNITS:

Acceleration	metre per second squared	...	m/s
activity (of a radioactive source)	disintegration per second	...	(disintegration)/s
angular acceleration	radian per second squared	...	rad/s
angular velocity	radian per second	...	rad/s
area	square metre	...	m
density	kilogram per cubic metre	...	kg/m
electric capacitance	farad	F	A·s/V
electrical conductance	siemens	S	A/V
electric field strength	volt per metre	...	V/m
electric inductance	henry	H	V·s/A
electric potential difference	volt	V	W/A
electric resistance	ohm		V/A
electromotive force	volt	V	W/A
energy	joule	J	N·m
entropy	joule per kelvin	...	J/K
force	newton	N	kg·m/s
frequency	hertz	Hz	(cycle)/s
illuminance	lux	lx	lm/m
luminance	candela per square metre	...	cd/m
luminous flux	lumen	lm	cd·sr
magnetic field strength	ampere per metre	...	A/m
magnetic flux	weber	Wb	V·s
magnetic flux density	tesla	T	Wb/m
magnetomotive force	ampere	A	...
power	watt	W	J/s
pressure	pascal	Pa	N/m
quantity of electricity	coulomb	C	A·s
quantity of heat	joule	J	N·m
radiant intensity	watt per steradian	...	W/sr
specific heat	joule per kilogram-kelvin	...	J/kg·K
stress	pascal	Pa	N/m
thermal conductivity	watt per metre-kelvin	...	W/m·K
velocity	metre per second	...	m/s
viscosity, dynamic	pascal-second	...	Pa·s
viscosity, kinematic	square metre per second	...	m/s
voltage	volt	V	W/A
volume	cubic metre	...	m
wavenumber	reciprocal metre	...	(wave)/m
work	joule	J	N·m

SI PREFIXES:

Multiplication Factors	Prefix	SI Symbol
$1\ 000\ 000\ 000\ 000 = 10^{12}$	tera	T
$1\ 000\ 000\ 000 = 10^{9}$	giga	G
$1\ 000\ 000 = 10^{6}$	mega	M
$1\ 000 = 10^{3}$	kilo	k
$100 = 10^{2}$	hecto*	h
$10 = 10^{1}$	deka*	da
$0.1 = 10^{-1}$	deci*	d
$0.01 = 10^{-2}$	centi*	c
$0.001 = 10^{-3}$	milli	m
$0.000\ 001 = 10^{-6}$	micro	μ
$0.000\ 000\ 001 = 10^{-9}$	nano	n
$0.000\ 000\ 000\ 001 = 10^{-12}$	pico	p
$0.000\ 000\ 000\ 000\ 001 = 10^{-15}$	femto	f
$0.000\ 000\ 000\ 000\ 000\ 001 = 10^{-18}$	atto	a

* To be avoided where possible.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

AMERICAN REVOLUTION BICENTENNIAL 1776-1976